

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

На правах рукопису
УДК 534.1+517.9

До захисту допущено
В. о. завідувача кафедри ММСА

О.Л.Тимощук

«___» _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра за спеціальністю 124 Системний аналіз
на тему: «Методи нелінійної оптимізації в задачах проектування електричних
двигунів загальнопромислового призначення»

Виконав:

студент II курсу, групи КА-71 мп
Курін Ілля Миколайович

Керівник: доцент кафедри ММСА,
к.т.н., доц. Тимощук О.Л.

Рецензент: доцент кафедри
електромеханіки
КПІ ім. Ігоря Сікорського,
к.т.н., доц. Чумак В.В.

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів
без відповідних посилань

Студент _____

Київ
2018

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Рівень вищої освіти — другий (магістерський)

Спеціальність (спеціалізація) — 124 «Системний аналіз» («Системний аналіз і управління»)

ЗАТВЕРДЖУЮ

В. о. завідувача кафедри ММСА

О. Л. Тимошук

« ____ » _____ 2018 р.

ЗАВДАННЯ

на магістерську дисертацію студенту Куріну Іллю Миколайовичу

1. Тема дисертації: «Методи нелінійної оптимізації в задачах проектування електричних двигунів загальнопромислового призначення», науковий керівник дисертації Тимошук Оксана Леонідівна, к.т.н., доцент, затверджені наказом по університету від «07» листопада 2018 р. № 4121-с

2. Термін подання студентом дисертації: _____

3. Об'єкт дослідження: оптимальне проектування електричних двигунів

4. Предмет дослідження: методи та алгоритми оптимального проектування електричних двигунів загальнопромислового призначення

5. Перелік завдань, які потрібно розробити:

- 1) провести аналіз методів нелінійної оптимізації для вирішення прикладних задач;
- 2) вибрати методи нелінійної оптимізації;
- 3) розробити алгоритм пошуку оптимальних електричних двигунів;
- 4) виконати програмну реалізацію системи пошуку оптимальних електричних двигунів;
- 5) виконати тестування програмної частини системи

6. Орієнтовний перелік графічного (ілюстративного) матеріалу:

- 1). Побудова моделі. Схема структурна
- 2). Обмеження на систему

- 3). Критерій оптимальності
- 4). Метод штрафних функцій. Схема алгоритму
- 5). Метод множників. Схема алгоритму
- 6). Метод комплексів. Схема алгоритму
- 7). Метод допустимих напрямків. Схема алгоритму
- 8). Інтерфейс програмного продукту
- 9). Результати проектування

7. Орієнтовний перелік публікацій:

(1) Чумак В.В. Оптимальне проектування однофазного конденсаторного асинхронного двигуна із використанням модифікованого методу множників Лагранжа /В.В.Чумак, О.Л.Тимощук, І.М. Курін //Гідроенергетика України. — 2017. — № 1—2. — С. 56—60

(2) Курін І.М. Методи нелінійної оптимізації в задачах проектування електричних двигунів /І.М.Курін, О.Л.Тимощук //Системні науки та кібернетика. — 2017. — № 6. — С. 136—148

8. Дата видачі завдання: _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Написання вступу магістерської дисертації	07.09.2018—13.09.2018	
2.	Порівняльний аналіз методів нелінійної оптимізації	14.09.2018—23.09.2018	
3.	Вибір методів нелінійної оптимізації	24.09.2018—30.09.2018	
4.	Підготовка матеріалів першого розділу магістерської дисертації	01.10.2018—08.10.2018	
5.	Розроблення алгоритму пошуку оптимальних електричних двигунів	09.10.2018—17.10.2018	
6.	Підготовка матеріалів другого розділу магістерської дисертації	18.10.2018—26.10.2018	
7.	Підготовка графічної частини магістерської дисертації	27.10.2018—04.11.2018	
8.	Підготовка матеріалів третього розділу магістерської дисертації	05.11.2018—14.11.2018	
9.	Підготовка матеріалів розділу стартап-проекту магістерської дисертації	15.11.2018—22.11.2018	
10.	Написання висновку магістерської дисертації	23.11.2018—26.11.2018	

Студент

І.М.Курін

Науковий керівник дисертації

О.Л.Тимощук

РЕФЕРАТ

Магістерська дисертація: 99 с., 58 рис., 24 табл., 1 додаток, 26 джерел.

Об'єктом дослідження є оптимальне проектування електричних двигунів.

Предметом дослідження є методи та алгоритми оптимального проектування електричних двигунів загальнопромислового призначення.

Мета дослідження:

а) дослідження числових алгоритмів та методів оптимізації для проектування електричних двигунів;

б) розробка електронного Інтернет-ресурсу, який знаходить оптимальні параметри для проектування однофазного асинхронного двигуна в режимі онлайн.

Теоретичною та методологічною основою дослідження є праці вітчизняних і зарубіжних вчених в галузі математичного моделювання, дослідження операцій, математичного програмування та прикладного оптимального проектування.

В ході магістерської дисертації створено сайт в мережі Інтернет для визначення декількома числовими методами нелінійної оптимізації параметрів однофазного асинхронного двигуна, при яких обраний користувачем критерій якості виробу досягає оптимальних значень та задовольняються потрібні технічні обмеження.

Методологія реалізована на основі уже відомих алгоритмів та з використанням власних розробок.

Програмний продукт реалізовано за допомогою мови програмування C# та програмної платформи .NET. Надано рекомендації до подальших досліджень.

ОПТИМАЛЬНЕ ПРОЕКТУВАННЯ, ОДНОФАЗНИЙ АСИНХРОННИЙ ДВИГУН, МЕТОДИ ОПТИМІЗАЦІЇ, МАТЕМАТИЧНЕ ПРОГРАМУВАННЯ, НЕЛІНІЙНА ОПТИМІЗАЦІЯ.

ABSTRACT

Master's thesis: 99 p., 58 fig., 24 tabl., 1 appendix, 26 sources.

The topic of the research: "Methods of nonlinear optimization in problems of design of electric engines for general industrial applications".

The object of research is optimal design of electric engines.

The subject of the study are methods and algorithms of optimal design of engines for general industrial applications.

The aim of the study:

- a) research numerical algorithms and methods for design electric engines;
- b) develop electronic Internet-resource that seek optimal parameters for design of single-phase asynchronous engine.

Theoretical and methodological basis of the study are works of domestic and foreign scholars in the field of math modeling, operations research, math programming and applied optimal design.

During the thesis created Internet site to determine the parameters of single-phase asynchronous engine, by which the user-selected quality criteria of product reach its optimum and needed technical limitations become satisfied, using several numerical methods of nonlinear optimization.

The methodology is implemented on the basis of already known algorithms and using own development.

The software is implemented using the programming language C# and framework .NET. The recommendations for further research are given.

OPTIMAL DESIGN, SINGLE-PHASE ASYNCHRONOUS ENGINE,
OPTIMIZATION METHODS, MATH PROGRAMMING, NONLINEAR
OPTIMIZATION.

ЗМІСТ

ПЕРЕЛІК ПРИЙНЯТИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ	8
ВСТУП.....	9
РОЗДІЛ 1 ЕЛЕМЕНТИ ПРОЕКТУВАННЯ ТЕХНІЧНИХ СИСТЕМ	12
1.1 Порівняння синтезу та аналізу в технічному проектуванні...	12
1.2 Концепція системи в техніці	14
1.3 Оптимальне проектування в теорії технічних систем	16
1.4 Конструкція і принцип роботи однофазного асинхронного двигуна	18
Висновки до розділу	23
РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ ОПТИМАЛЬНОГО ПРОЕКТУВАННЯ ДВИГУНІВ.....	24
2.1 Побудова моделі	24
2.2 Критерії оптимальності в задачах з обмеженнями.....	26
2.3 Метод штрафних функцій	28
2.4 Метод множників	32
2.5 Метод комплексів	34
2.6 Метод допустимих напрямків.....	38
Висновки до розділу	41
РОЗДІЛ 3 АНАЛІЗ МЕТОДІВ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ ТА ЇХ ПРАКТИЧНЕ ЗАСТОСУВАННЯ ПРИ ОПТИМАЛЬНОМУ ПРОЕКТУВАННІ ОДНОФАЗНОГО АСИНХРОННОГО ДВИГУНА ІЗ ЗАСТОСУВАННЯМ СТВОРЕНОГО ІНТЕРНЕТ-РЕСУРСУ	43
3.1 Порівняння методів нелінійного програмування.....	43
3.2 Інтерфейс онлайн-ресурсу	50
3.3 Робота з онлайн-ресурсом.....	57
3.4 Пошук допустимої точки	63
3.5 Оптимальне проектування однофазного асинхронного двигуна з використанням онлайн-ресурсу	66
Висновки до розділу	77

РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	78
4.1 ОПИС ІДЕЇ ПРОЕКТУ	78
4.2 ТЕХНОЛОГІЧНИЙ АУДИТ ІДЕЇ ПРОЕКТУ	80
4.3 АНАЛІЗ РИНКОВИХ МОЖЛИВОСТЕЙ ЗАПУСКУ СТАРТАП-ПРОЕКТУ	81
4.4 РОЗРОБЛЕННЯ РИНКОВОЇ СТРАТЕГІЇ ПРОЕКТУ	89
4.5 РОЗРОБЛЕННЯ МАРКЕТИНГОВОЇ ПРОГРАМИ СТАРТАП-ПРОЕКТУ	91
Висновки до розділу	94
ВИСНОВКИ ПО РОБОТІ І РЕКОМЕНДАЦІЇ ДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ	96
ПЕРЕЛІК ПОСИЛАНЬ	97
ДОДАТОК А ЛІСТИНГ ПРОГРАМИ	100

ПЕРЕЛІК ПРИЙНЯТИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ККД – коефіцієнт корисної дії

ЕОМ – електронна обчислювальна машина

МРС – магніторушійна сила

ОАД – однофазний асинхронний двигун

ЗНП – задача нелінійного програмування

ВСТУП

Науково-технічний прогрес в різних галузях промисловості викликає зростання потреб в електричних машинах малої потужності. Як показує світовий досвід, виробництво мікромашин представляє собою самостійну галузь, яка динамічно розвивається, і темпи зростання якої перевищують темпи зростання електромашинобудівництва в цілому.

Розробка нових технологічних процесів виробництва електричних машин відкриває широкі можливості для підвищення продуктивності праці та ефективності виробничої діяльності. Виробництво електричних машин малої потужності характеризується багатомоделлюваністю, різноманітністю конструкцій, підвищеними вимогами до надійності та технічного рівня, технологічності конструкцій, підвищенням рівня спеціалізації та кооперації виробництва.

Технологічність конструкції надає істотного впливу на техніко-економічні показники серійного та масового виробництва електричних машин малої потужності. Конструкція спроектованої електричної машини має бути технологічною, тобто новий двигун має мати певну з технологічних міркувань величину повітряного зазору, коефіцієнт заповнення пазу, ширину зубців, відношення довжини активної зони до зовнішнього діаметру тощо.

Сучасні електричні машини є досить проблемними у тепловому відношенні. Тепловий розрахунок дозволяє контролювати вибір електромагнітних навантажень, забезпечуючи високе використання активних матеріалів без зниження надійності двигуна.

Проблема створення електричних машин, які мають оптимальне відношення електромагнітних навантажень та геометричних розмірів, з'явилася із зародженням електромашинобудівництва і залишається актуальною до сьогодення часу. Праці багатьох вчених були присвячені пошукам закономірностей, які б дозволили із меншими витратами праці та часу вирішити цю проблему відповідно до

конкретно заданих умов. Проте необхідність врахування великої кількості факторів, які є часто суперечливими, ускладнює отримання та використання таких закономірностей при вирішенні прикладних задач.

Оптимізація проекту є головною ціллю кожного інженера, який намагається створити окремий елемент, пристрій чи систему для забезпечення певних потреб. Проте здійснення цієї цілі зазвичай є досить проблематичним, адже у розпорядженні конструкторів знаходиться невелика кількість стандартизованих методів, здатних допомогти в оптимізації.

Поява швидкодіючих ЕОМ відкрила нові можливості для дослідження та оптимального проектування електричних машин. Програми оптимального проектування використовувались при створенні серій двигунів А2 і А4 та їх модифікацій, при розробці вибухобезпечних двигунів тощо. Роботи по оптимізації двигунів проводились в інституті електродинаміки НАН України. У перспективі – створення автоматизованих систем проектування електричних машин. Задача оптимізації електричних машин зводиться до пошуку локального, а іноді, глобального, чи достатньо близького до нього екстремуму функції якості в багатовимірному просторі із нелінійною системою обмежень. В даній роботі розглядаються такі градієнтні методи нелінійної оптимізації, як метод штрафних функцій, метод множників та метод допустимих напрямків, а також один з методів прямого пошуку, а саме, метод комплексів.

Вибір критерію якості залежить від призначення електричної машини і пред'явлених до неї вимог. Для деяких електричних машин доцільно вибрати мінімум маси чи мінімум приведених витрат, для інших максимум енергетичних показників – ККД, $\cos \varphi$, іноді визначними критеріями є перенавантажувальна здатність – K_m і підвищені пускові властивості – K_p . Таким чином, неможливо знайти універсальний критерій оптимальності. В даній роботі було досліджено створення електричного двигуна за умови, що інженер сам обирає критерій оптимальності, тобто у нього є можливість власноруч задати необхідну функцію в якості критерію

серед декількох технічних характеристик двигуна в залежності від специфіки потрібної електричної машини.

Ціль оптимального проектування – отримати електричну машину, яка, задовольняючи усім вимогам системи обмежень по ваговим та енергетичним показникам, відповідає технічним та технологічним вимогам.

На основі вищевикладеного, ціль магістерської дисертації формулюється наступним чином: оптимальне проектування електричного двигуна і порівняння приведених методів оптимізації щодо швидкодії та комп'ютерної пам'яті, на прикладі конденсаторного двигуна потужністю 250 Вт, частота обертання 1500 об/хв.

РОЗДІЛ 1 ЕЛЕМЕНТИ ПРОЕКТУВАННЯ ТЕХНІЧНИХ СИСТЕМ

1.1 Порівняння синтезу та аналізу в технічному проектуванні

В багатьох працях вчених техніка визначається як «мистецтво чи наука практичного перетворення в практичних застосуваннях знань в областях чистих наук, таких, як фізика, хімія, біологія і т.д.». Хоча це визначення і загальне, воно передбачає, що ціллю техніки є синтез, чи зведення воедино, корисних для практики систем шляхом застосування знань і методів, отриманих із «чистих» наук. Значення слова «практичний» в цьому визначенні слідую трактувати як найкращий чи оптимальний. Це означає, що ціллю технічного проектування є розробка найкращої можливої системи для даного застосування відповідно до ресурсів, виділених на проект.

Головним фактором того, щоб оптимізація конструкцій стала можливою, є наявність швидкодіючих цифрових обчислювальних машин. Оскільки таку машину можна розглядати як пристрій, який має справу з великою кількістю даних і який швидко виконує алгебраїчні та логічні операції, важливо розглянути спочатку роль обчислень у проектуванні. Звичайна практика, коли спочатку виробляється зразок для дослідів і визначається його працездатність, була доповнена додатковим етапом попереднього проектування, на якому для дослідження системи використовуються математичні моделі та числовий аналіз, безпосередньо до коштовного етапу розробки моделі з матеріалів. Наприклад, при проектуванні конструкції можна обрати конфігурації і розміри частин, а потім провести математичний дослід конструкції, розглядаючи її реакцію на певне навантаження. Якщо конструкція не веде себе так, як треба, то проводиться її заміна та виконується повторний дослід. Даний процес продовжується до тих пір, поки проектувальник не буде задоволений отриманим проектом.

Взагалі кажучи, проектувальник, перед тим як впевнитись, що він отримав найкращу систему, повинен мати можливість дослідити всі кандидатури на цю роль.

У другій половині минулого сторіччя було здійснено істотне зростання у вивченні конструювання. Цифрові ЕОМ дозволили інженеру-конструктору проводити чисельні вивчення поведінки систем, які в минулому вивчалися лише якісно. Даний прогрес в області можливостей аналізу приніс величезну користь механіці та науці про конструювання. Аналіз конструкцій, дослідження напруг, аналіз механізмів, дослідження перенесення тепла - ось лише деякі напрями, в яких зроблено ефективне просування за останні 50 років.

До 60-х років минулого сторіччя в дослідженнях по конструюванню увага приділялася перш за все розвитку можливостей аналізу. Протягом цього періоду розвитку можливостей синтезу при проектуванні приділялася мала увага. В деяких областях механіки ця задача постає особливо гостро. Наприклад, в механіці конструкцій можна провести аналіз конструкції, на яку здійснюється певне навантаження, і отримати точні значення навантаження, зсуву і навіть власної частоти. Проте при цьому буде невідомо, як має бути спланована конструкція та які у неї повинні бути пропорції, щоб ефективно використовувати матеріал для забезпечення міцності.

Припускається, що можливості аналізу, необхідні для проектування з використанням ЕОМ, наразі є доступними. Наступним завданням, до якого потрібно перейти, є питання про те, як розуміти терміни «найкращий» чи «оптимальний». В різних сферах промисловості, які орієнтовані на прибуток, ціллю проектування є максимізація певної функції прибутку при обмеженнях на ресурси, кількість та умови праці людей. Як тільки було обрано деяку функцію і визначені обмеження, то проектувальнику системи треба мати метод, за допомогою якого він зможе отримати оптимальні проекти.

Надалі в даній роботі буде детально розглянуто вирішення всіх цих задач та надано рекомендації для створення однієї з технічних систем, а саме, однофазного конденсаторного двигуна.

1.2 Концепція системи в техніці

У середині 50-х років минулого сторіччя в літературі почав з'являтися формалізований підхід до розробки багатовимірних систем, створюваних людиною [1, 2]. Цей метод дослідження має особливості, загальні для більшості процедур розв'язання задач; він був названий «технічною розробкою системи». Характерною особливістю, яка віддаляє розробку системи і проектування з використанням ЕОМ від більшості логічних схем вирішення задач, є увага, яка приділяється кількісному опису системи та її поведінки. Основна ідея технічної розробки системи полягає в тому, щоб, почавши з формулювання потреб і цілей системи, просуватися деяким регулярним чином до оптимальної системи. Спрощена модель процесу розробки системи, яка показана на рисунку 1.1, показує, що оптимальне проектування є частиною технічної розробки системи, але жодним чином не є головною частиною.

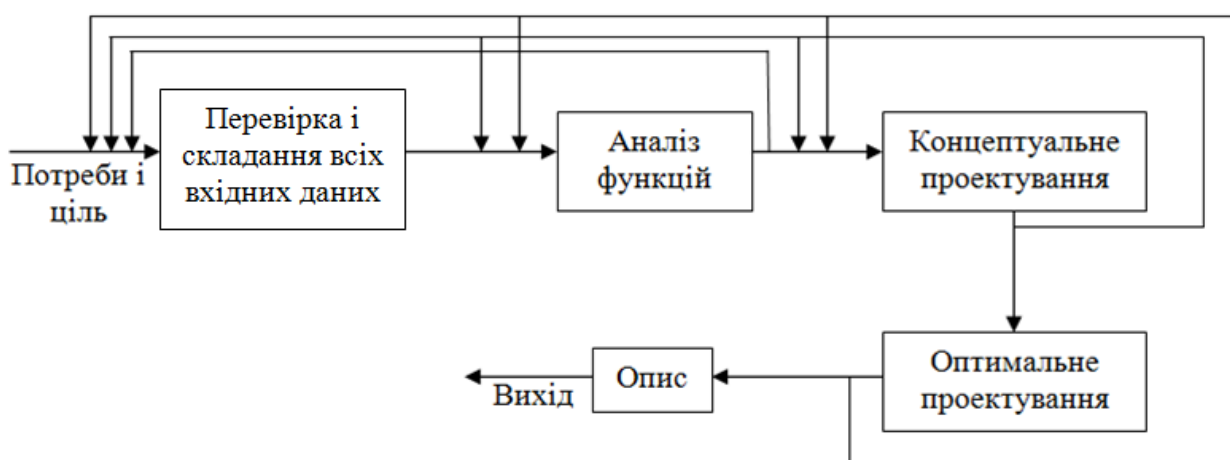


Рисунок 1.1 – Модель технічної розробки системи

Технічна розробка системи починається з визначення потреб потенційного користувача системи, яка розробляється. Часто виявляється, що користувач бажає бачити систему, яка виконує дану роботу, проте не може кількісно сформулювати свої потреби і цілі. Тому у інженера та користувача виникає додаткова відповідальність щодо кількісного втілення цілей системи так, щоб була визначена змістовна сукупність цільових функцій для наступної розробки.

Як тільки потреби та цілі системи визначені, необхідно дати кількісний опис функцій, які має виконувати система і будь-яка потрібна підсистема. Ця процедура називається аналізом функцій. Його ціль полягає у виборі функцій, чи операцій, які мають бути приведені для виконання задачі, яку має виконувати майбутня система. Визначення функцій за своєю суттю має скоріше якісний характер. Проти, як тільки функція, чи операція, визначена, вона має бути описана в кількісних термінах. Наприклад, якщо операція має бути проведена швидко, необхідно оговорити допустимий час.

Наступний крок, показаний на рисунку 1.1, є одним з найскладніших етапів в технічній розробці системи і, звісно, найскладнішим кроком для аналітичного опису. Концептуальне проектування, як видно з назви, полягає у визначенні концепцій, чи основних конфігурацій системи, які можуть задовольнити цілі системи. На цьому етапі процесу проектування важливо визначити межі значень параметрів, які описують систему. Для параметрів проектування в межах цих значень система має вміти виконувати функції, які визначені на попередньому кроці.

Нехай тепер маємо справу із оптимальним проектом, для якого ціллю є вибір невизначених параметрів, введених на попередньому кроці. Ці параметри мають бути в межах, визначеними технологічними обмеженнями і призначенням системи. Критерієм вибору параметрів системи є максимізація коштовності системи чи мінімізація міри затрат. Треба зауважити, що математично точний оптимум може виявитися недосяжним і, як наслідок, може слугувати лише орієнтиром. Проте методи вибору параметрів системи повинні мати таку властивість, що якщо оптимум існує, то при достатньому терпінні та заданому машинному часі оптимум повинен бути досяжним у границі.

Те, що є останнім кроком в моделі технічної розробки системи на рисунку 1.1, – опис оптимальної системи – в дійсності виявляється проміжним кроком. До тих пір, поки процедура проектування системи не стане ефективною, система, запропонована групою розробників, ймовірно, не буде задовольняти замовника.

Отримавши результати одного проходження через процедуру проектування системи, замовник, можливо, згадає деякі обмеження, які він забув описати і які оптимальна система порушує. Крім того, проектувальник може також придумати привабливі ідеї, про які раніше не міркував. Він згадає технологічні обмеження, які він забув описати і які оптимальна система порушує. Врешті решт, замовник безумовно вирішить, що непогано було б зменшити на маленьку величину рівні якості системи, якщо це призведе до економії грошей.

На наступному кроці процедури кожен учасник групи повернеться до праці, маючи здобутий досвід. Тому ітеративна процедура, яка зображена на рисунку 1.1, буде продовжуватися до тих пір, поки замовник не вирішить, що запропонована система є такою, яку він справді потребує [3, 4].

1.3 Оптимальне проектування в теорії технічних систем

Теорія оптимального керування загалом пов'язана із знаходженням керування по оберненому зв'язку. Оптимально керована система має активні елементи, які відчують помилки на виході, які виникають через випадкові зміни у вхідних даних, і змінюють керовану дію системи із ціллю максимізації деякого критерію якості системи. З іншої сторони, в оптимальному проектуванні визначають елементи системи, чи параметри, які описують ці елементи, причому останні фіксовані впродовж усього часу існування елементів, так що система в деякому сенсі є оптимальною. В літературі по керуванню подібна ситуація часто називається керування із розімкненим ланцюгом [1, 5]. Принципова різниця в цих двох задачах полягає у тому, що в задачі оптимального проектування обрані значення змінних фіксовані протягом часу життя системи, в той час як при керуванні з оберненим зв'язком значення змінних змінюються в процесі роботи системи.

В більшості публікацій по оптимальному керуванню розглядаються динамічні системи [6, 7]. Це означає, що функціональні системи, які описуються задачею Коші, розвиваються у часі. Багато задач оптимального проектування не відносяться до цього типу. Наприклад, при проектуванні конструкції необхідно мати справу із напругами, які обумовлені прикладеними навантаженнями. Ці напруги визначаються за допомогою граничної задачі для зсувів, яка не може трактуватися як динамічний процес (задача Коші). В деяких задачах проектування можливо так визначити допоміжні змінні, що керування задачі зводяться до задачі Коші з додатковими обмеженнями. Проте ця процедура може дуже ускладнити задачу. З цієї причини система, яка розглядається в даній роботі, формується безпосередньо у вигляді граничної задачі.

Також слідє відмітити особливу різницю обчислювального характеру між двома класами задач проектування. До першого класу відносяться задачі, в яких проектування і функціонування системи описується скінченним числом змінних проектування, а до другого – системи, які описуються функціями, визначеними в деякій заданій області простору. Математично ці задачі трактуються як скінченновимірні та нескінченновимірні відповідно. Теорії оптимізації для цих двох класів задач може бути надана однакова форма, проте існує істотна різниця в обчислювальних методиках, прийнятних для оптимізації проекту.

Врешті решт, важливо відмітити те, що оптимізація технічного проекту та аналіз технічної розробки істотно різняться за своєю суттю. В аналізі необхідно впевнитись, що розв'язок існує, а числові методи стійкі [8, 9]. З іншої сторони, при оптимальному проектуванні не гарантовано навіть існування номінального проекту, який задовольняє обмеження, а також існування оптимального проекту. Якщо навіть оптимальний проект існує, то числові методи його побудови часто виявляються досить чутливими до початкових значень, і для збіжності ітерацій треба робити точні обчислення.

1.4 Конструкція і принцип роботи однофазного асинхронного двигуна

Однофазний асинхронний двигун - малопотужний механізм до 10 кВт. Однак завдяки своїй компактності і особливостями дії, його використання дуже велике. Найширша сфера його застосування - це побутові прилади з однофазним струмом. Однофазні асинхронні електродвигуни застосовуються для холодильників, центрифуг, пральних машин, часто використовується для малопотужних вентиляторів.

Розрізняють біфілярний і конденсаторний механізм роботи однофазного асинхронного двигуна:

а) при біфілярному пуску обмотка не використовується при постійному режимі, інакше значення ККД знижується. Набираючи обертів, вона обривається. Обмотка пуску вмикається на кілька секунд. Розрахунок роботи за 3 секунди до 30 разів на 60 хвилин. Перевищення запусків може призвести до перегріву витків;

б) при конденсаторному пуску фаза розщеплена, ланцюг допоміжної обмотки включається під час запуску. Для досягнення пускового моменту необхідно створити кругове магнітне поле. Використання конденсатора забезпечує кращий пусковий момент. У конденсаторного пристрою дві котушки, які завжди під напругою [10].

В даній роботі досліджено саме однофазний асинхронний двигун з конденсаторним пуском, а потім продемонстровано математично обґрунтований метод покращення його технічних та технологічних характеристик.

Асинхронний конденсаторний двигун має на статорі дві обмотки (головну та допоміжну), які зсунуті у просторі на електричний кут $\pi/2$. Фази обмотки з'єднані паралельно. Як показано на рисунку 1.2, в одну з них (в допоміжну) включений робочий конденсатор C_p , який працює і при пуску, якщо потреби до пускового моменту не занадто великі. При завищених потребах до пуску тільки на деякий час пуску включається ще й пусковий конденсатор C_n паралельно працюючому

конденсатору C_p . В даному проекті такі умови не потребуються. В якості конденсатора було використано метало-паперовий герметизований частотний конденсатор марки МБГЧ-1 чи К-75 ємністю 10 мкФ.

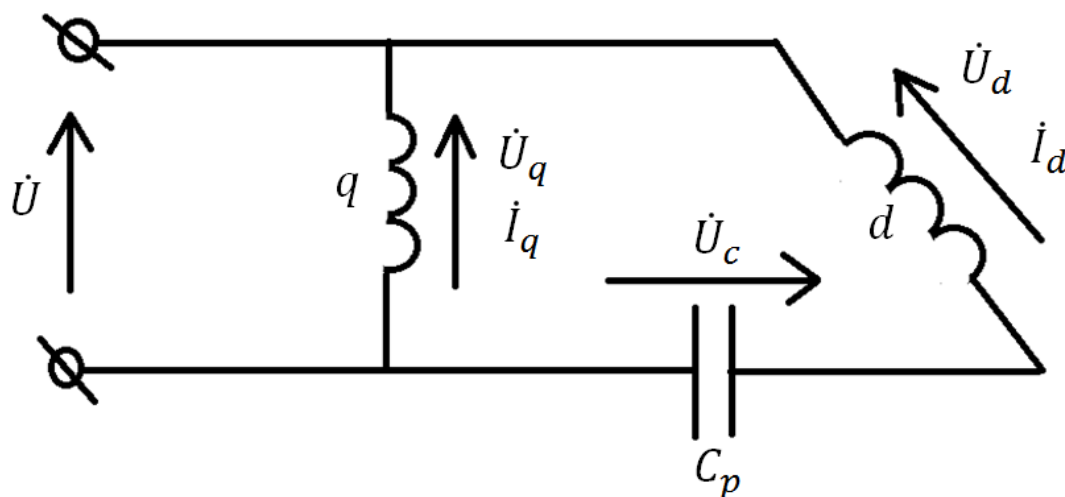


Рисунок 1.2 – Схема роботи однофазного асинхронного двигуна

Конденсаторний двигун по відношенню до мережі є однофазним, а по конструкції – двофазним. На відміну від двофазного двигуна конденсаторний двигун, як правило, має різну кількість витків обох обмоток, різні опори, перерізи обмоточних проводів цих обмоток.

Розглянемо загальну конструкцію асинхронного конденсаторного двигуна. Сердечник статора набирається з листів гарячекатаної ізотропної електротехнічної сталі марки 1211, товщиною 0,5 мм, із низьким вмістом кремнію, що дозволяє отримувати малі пази складної конфігурації. Листи зварюють електрозваркою в пакет і обробляють по зовнішньому діаметру. Пакет статора закріплюється безпосередньо між підшипниковими щитами за допомогою гайок та шайб. Підшипниковий щит виготовляється з алюмінію шляхом виливки під тиском, в якому є отвір для інтенсивного охолодження лобових частин обмотки статора потоком повітря. На одному з підшипникових щитів укріплена ізолююча втулка, через яку проведений кабель живлення. Обмотка статора є одношаровою. Складається з м'яких секцій, які намотані круглим проводом марки ПЗВ, секції закладаються в напівзакриті пази [10, 11].

Сердечник ротора виготовляється з тої самої сталі, що й статора, збирається на відправку, пресується й заливається під тиском алюмінієм. Одночасно відливаються короткозамикні кільця та вентиляційні лопатки. При обертанні ротора, вентиляційні лопатки викидають повітря через бокові отвори в підшипникових щитах. Повітря засмоктується через отвори і проганяється через лобові частини обмотки статора.

Сердечник ротора кріпиться на валу, виготовляється з сталі 45, вал пресується в пакет по посадці А/Гр, а потім виконується шліфування його і зовнішньої поверхні ротора з відносним биттям не більше, ніж 0,04 мм.

Підшипники хитання, які знаходяться в підшипникових щитах, захищають підшипники від забруднення. Зі сторони робочого кінця валу підшипники прижимаються до валу шайбою, яка запобігає різким повздовжнім зсувам ротора чи різкій зміні моменту на валу.

Кріплення двигуна всередині виробу виготовляються за допомогою фланцю, який виконаний як одне ціле з підшипниковим щитом.

На підшипниковий щит кріпиться табличка, на якій приведені паспортні дані двигуна. Маркування має бути нанесено на паспортній табличці двигуна фотохімічним чи іншим способом, який забезпечує ясність та збереження написів протягом усього часу експлуатації двигуна.

Однофазним на практиці прийнято називати двигун, який живиться від однофазних мереж змінного струму. Ці двигуни, як правило, за своїм устроєм є двофазними. Двигун живиться від однофазних мереж і обидві обмотки знаходяться постійно під напругою: і при пуску, і при роботі, тобто двигун включається і працює як двофазний. Двигун працює в несиметричному режимі (несиметричний за виконанням), тобто не при кругових, а при еліптичних обертальних полях.

В будь-якому несиметричному за виконанням двигуні може бути отримане кругове обертальне поле у повітряному зазорі, якщо на його зажими подати певним чином підібрану систему напруг. В цьому випадку ротор буде вести себе так само, як і симетричному режимі, тобто буде мати симетричну систему струмів. Наявність симетричної системи струмів в роторі пояснюється тим, що на нього діє лише

магнітне поле, створене статором. При цьому не має різниці, яким чином і якою системою створене поле, а важливим є лише характер діючого поля.

Змінний струм, який протікає по одній фазі статора створює пульсуючу МРС, яку можна розкласти на дві МРС, які обертаються у протилежні сторони із однаковою частотою обертання ($n_1 = f_1/p$) [12]. Амплітуди МРС рівні між собою і складають половину амплітуди пульсуючої МРС. МРС створюють магнітне поле. Одне поле умовно називають прямим, воно напрямлене у сторону обертання ротора, друге – оберненим, воно напрямлене проти обертання ротора.

Частота обертання ротора відносно прямого поля

$$(n_1 - n_2) = S = 1 - \omega, \quad (1.1)$$

а відносно оберненого поля

$$(n_1 + n_2) = 2 - S = 1 + \omega. \quad (1.2)$$

При пуску, коли $S=1$ ($n=0$), ці поля створюють однакові, але різні за знаком моменти M_1 і M_2 , тому результуючий момент двигуна M_3 при пуску рівний нулю, як показано на рисунку 1.3. Якщо ротор привести до обертання в ту чи іншу сторону, то один з моментів (M_1 чи M_2) буде переважати і при умові, що результуючий електромагнітний момент більше статичного ($M_3 > M_c$), двигун досягне певної усталеної частоти обертання. Відповідно до схем заміщення можна отримати результуючий момент ($M_3 = M_1 - M_2$).

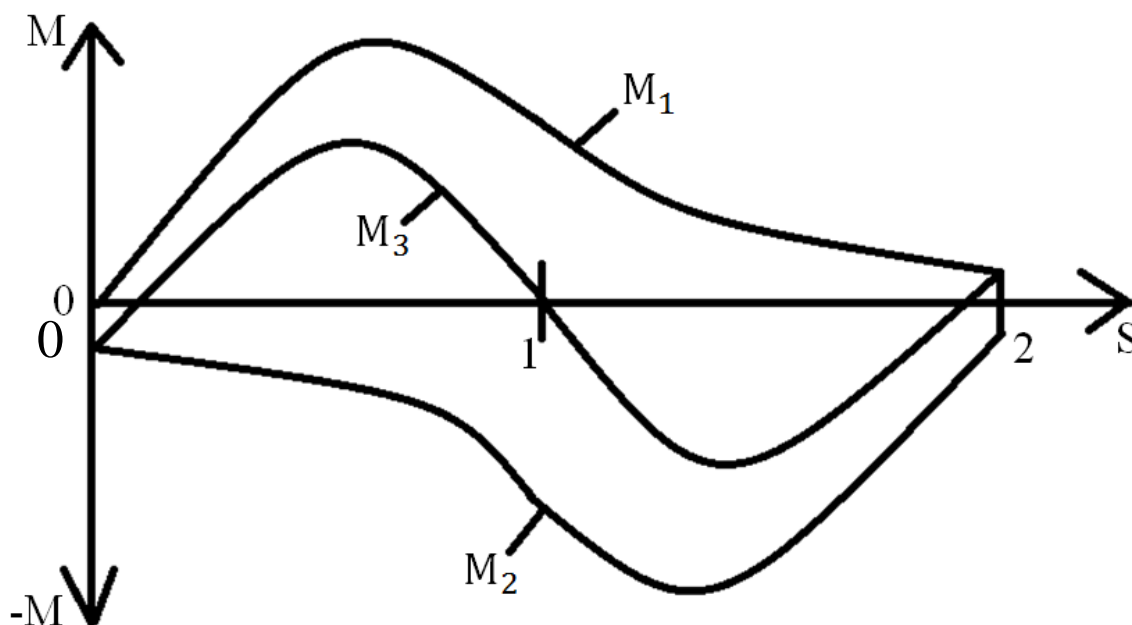


Рисунок 1.3 – Залежність моментів двигуна

Асинхронний конденсаторний двигун має на статорі дві обмотки, які є робочими. В одну з них включається робочий конденсатор C_p .

Якщо ємність конденсатора обрана таким чином, що при номінальному навантаженні у повітряному зазорі створюється кругове поле, то конденсаторний двигун при інших рівних умовах має найбільший ККД і невисоку кратність пускового моменту $K_n \approx 0,25 \div 0,3$. Для того, щоб підняти кратність пускового моменту до $K_n \approx 0,4 \div 0,55$, змінюють обмоточні дані та ємність робочого конденсатора. В цьому випадку в номінальному робочому режимі у повітряному зазорі створюється еліптичне поле, що призводить до деякого падіння ККД. Якщо конденсаторний двигун повинен мати кратність пускового моменту $K_n > 0,55$, то використовують двигун з робочим і пусковим конденсатором (чи іншим пусковим фазозсувним елементом). В цьому випадку ємність робочого конденсатора обирають з умови створення кругового поля при номінальному навантаженні, а параметри фазозсувного елементу (з урахуванням ємності робочого конденсатора) – з умови створення потрібного пускового моменту.

Висновки до розділу

У розділі розглянуто елементи проектування технічних систем та приведено основні факти з даної теми.

Показано етапи оптимального проектування технічних систем та розглянуто основні проблеми, які можуть виникнути в процесі розробки технічного пристрою. Після цього була продемонстрована відмінність між оптимальним проектування статичних та динамічних технічних систем.

В кінці розділу було розглянуто будову та принцип роботи однофазного асинхронного двигуна, який буде спроектовано в наступних розділах даної роботи.

РОЗДІЛ 2 МАТЕМАТИЧНІ МЕТОДИ ОПТИМАЛЬНОГО ПРОЕКТУВАННЯ ДВИГУНІВ

2.1 Побудова моделі

Оптимізація електричних машин є окремим випадком багатопараметричного оптимального проектування. Об'єкт оптимізації – електричну машину – можна розглядати як систему функцій від змінних входніх параметрів X_1, X_2, \dots, X_n і вихідної величини F . В загальному випадку така система має n змінних параметрів, які називаються незалежними змінними. Входом системи є n -мірний вектор $\bar{X} = (X_1, X_2, \dots, X_n)^T$, а вихід системи – функціонально залежна від стану незалежних змінних скалярна величина $F(\bar{X})$. Величина $F(\bar{X})$, характеризуюча степінь якості машини, яка проектується, отримала назву критерія оптимальності чи функції якості.

Як на входні параметри, так і на характеристики об'єкта оптимізації можуть бути накладені обмеження, які обумовлені потребами стандартів, технічних умов та інших нормативних документів. Ці обмеження, які залежать від вектора \bar{X} , можливо привести до виду $G_j(\bar{X}) \geq 0$, де $j=1, 2, \dots, m$.

Задача оптимального проектування – відшукати із заданою точністю в області існування об'єкта, який оптимізується, вектор $\bar{X}^* = (X_1^*, X_2^*, \dots, X_n^*)$, який відповідає найкращому (мінімальному чи максимальному) значенню функції якості.

При проектуванні ОАД доцільно обмежитись сімома незалежними змінними: діаметром розточення статора $D_1 (X_1)$, довжиною пакета статора $l_1 (X_2)$, шириною зубця статора $b_{zs} (X_3)$, шириною зубця ротора $b_{zr} (X_4)$, висотою стінки статора $h_{as} (X_5)$, висотою стінки ротора $h_{ar} (X_6)$ та зовнішнім діаметром листа статора $D_a (X_7)$ [11, 13].

Контрольовані характеристики ОАД – це регламентовані стандартами чи технічними умовами значення кратності пускового початкового моменту $K_n (G_1)$, кратності пускового початкового струму $K_i (G_2)$, кратності максимального моменту

K_{\max} (G_3), перевищення температури обмотки статора $T_{об}$ (G_4), коефіцієнта заповнення пази $K_{зап}$ (G_5), коефіцієнта корисної дії ККД (G_6), коефіцієнта потужності $\cos \varphi$ (G_7) та маси електротехнічної сталі M (G_8) [14, 15]. Розглянемо задачу, коли критерієм оптимальності двигуна взято масу електротехнічної сталі, і позначимо дану характеристику літерою F , а інші характеристики двигуна приймемо за обмеження задачі.

Маса електротехнічної сталі є сумою мас сталі спинок та зубців статора і ротора:

$$F = GAS + GAR + GZS + GZR \quad (2.1)$$

де

а) GAS – маса сталі спинки статора, кг:

$$GAS = \gamma_c \pi (D_a - h_{as}) h_{as} l_l K_c \quad (2.2)$$

б) GAR – маса сталі спинки ротора, кг:

$$GAR = \gamma_c \pi (d_b + h_{ar}) h_{ar} l_l K_c \quad (2.3)$$

в) GZS – маса сталі зубців статора, кг:

$$GZS = \gamma_c Z_c b_{zs} h_{ns} l_l K_c \quad (2.4)$$

г) GZR – маса сталі зубців ротора, кг:

$$GZR = \gamma_c Z_c b_{zr} h_{nr} l_l K_c, \quad (2.5)$$

де $\gamma_c = 7800 \frac{\text{кг}}{\text{м}^3}$ – питома маса сталі;

$K_c=0.95$ – коефіцієнт заповнення пакета сталлю;

$d_B = 0.02$ м – діаметр вала;

$Z_c = 24$ – кількість пазів статора;

h_{ns} та h_{nr} – проміжні змінні:

$$\begin{aligned} h_{ns} &= \frac{(D_a - D_l - 2 * h_{as})}{2} \\ h_{nr} &= \frac{\pi(D_l - 0.0019) - 17 * b_{zr}}{17 + \pi} + 0.0007 \end{aligned} \quad (2.6)$$

Таким чином, була отримана така задача нелінійної оптимізації:

$$\begin{cases} F(\bar{X}) \rightarrow \min \\ \tilde{G}_j(\bar{X}) \geq 0, j = 1, \dots, m \\ X_i \geq 0, i = 1, \dots, n \end{cases} \quad (2.7)$$

де $m=n=7$;

$\tilde{G}_j(\bar{X})$ – видозмінені функції обмежень: $\tilde{G}_j(\bar{X}) = G_j(\bar{X}) - b_j$ або $\tilde{G}_j(\bar{X}) = b_j - G_j(\bar{X})$;

b_j – обмеження на технічні параметри двигуна.

2.2 Критерії оптимальності в задачах з обмеженнями

Розглянемо наступну ЗНП з обмеженнями у виді нерівностей:

$$\begin{cases} f(\bar{x}) \rightarrow \min \\ g_j(\bar{x}) \geq 0, j = 1, \dots, m \end{cases} \quad (2.8)$$

Обмеження у виді нерівності $g_j(\bar{x}) \geq 0$ називається активним в точці \bar{x} , якщо $g_j(\bar{x}) = 0$, і неактивним, якщо $g_j(\bar{x}) > 0$.

Якщо існує можливість знайти обмеження, які неактивні в точці оптимуму, до безпосереднього розв'язання задачі, то ці обмеження можна виключити з моделі і тим самим зменшити її розмірність. Основна складність полягає при цьому в ідентифікації неактивних обмежень, яка б передувала розв'язанню задачі.

Кун і Такер побудували необхідні та достатні умови оптимальності для задач нелінійного програмування за припущенням, що f і g_j – диференційовані функції [16, 17]. Ці умови оптимальності, які відомі під назвою умови Куна-Такера, можна сформулювати у виді задачі знаходження розв'язку деякої системи нелінійних рівнянь і нерівностей, або, як іноді кажуть, задачі Куна-Такера.

Для задачі нелінійного програмування із обмеженнями у виді нерівностей, задача Куна-Такера виглядає так:

$$\begin{cases} \nabla f(x) - \sum_{j=1}^m u_j \nabla g_j(x) = 0 \\ g_j(x) \geq 0, j = 1, \dots, m \\ u_j g_j(x) = 0, j = 1, \dots, m \\ u_j \geq 0, j = 1, \dots, m \end{cases} \quad (2.9)$$

Тепер розглянемо строгі формулювання необхідних і достатніх умов оптимальності розв'язку задачі нелінійного програмування:

а) нехай f і g_j – диференційовані функції, а x^* – допустимий розв'язок даної задачі. Покладемо $I = \{j | g_j(x^*) = 0\}$. Далі нехай $\nabla g_j(x^*)$ при $j \in I$ лінійно незалежні. Якщо x^* – оптимальний розв'язок задачі нелінійного програмування, то існує такий вектор u^* , що (x^*, u^*) є розв'язком задачі Куна-Такера (необхідна умова). Слід зауважити, задача Куна-Такера може не мати розв'язок, якщо не виконується умова лінійної незалежності в точці оптимуму;

б) нехай цільова функція f – опукла, усі обмеження у виді нерівностей – увігнуті функції g_j , $j=1, \dots, m$. Тоді, якщо існує розв'язок (x^*, u^*) , який задовольняє

умовам Куна-Такера, то x^* - оптимальний розв'язок задачі нелінійного програмування (достатня умова).

Для практичних задач умова лінійної незалежності, як правило, виконується. Якщо в задачі всі функції диференційовані, то точку Куна-Такера можна розглядати як можливу точку оптимуму. Таким чином, багато методів нелінійного програмування збігаються саме до точки Куна-Такера.

2.3 Метод штрафних функцій

Почнемо огляд методів вирішення ЗНП з обмеженнями у виді нерівностей (2.9). За допомогою побудованих далі алгоритмів в просторі R^n будується скінченна послідовність точок $x^t, t = 0, 1, \dots, T$, яка починається із заданої точки x^0 і закінчується точкою x^T , яка дає найкраще наближення до x^* серед усіх точок побудованої послідовності.

В методі штрафних функцій в якості $x^t, t = 1, \dots, T$ беруться стаціонарні точки так званої штрафної функції – цільової функції допоміжної задачі безумовної оптимізації. За допомогою штрафної функції вихідна задача умовної мінімізації перетворюється в послідовність задач безумовної мінімізації [18].

Штрафна функція визначається виразом

$$P(x, R) = f(x) + Q(R, g(x)), \quad (2.10)$$

де R – набір штрафних параметрів;

так званий штраф Q є функцією R і функцій, які задають обмеження.

Q визначається так, щоб допустимі точки задачі мали перевагу перед недопустимими в процесі безумовної мінімізації штрафної функції.

Розглянемо деякі види штрафів для обмежень у вигляді нерівностей [16, 19]:

а) так званий, нескінченний бар'єр, який показано на рисунку 2.1 і який можна описати недиференційованою функцією:

$$Q(g(x)) = \begin{cases} 0, & g(x) \geq 0 \\ +\infty, & g(x) < 0 \end{cases} \quad (2.11)$$

Машина реалізація нескінченного бар'єру неможлива, проте можна використати досить велике число, яке допускає запис в ЕОМ;

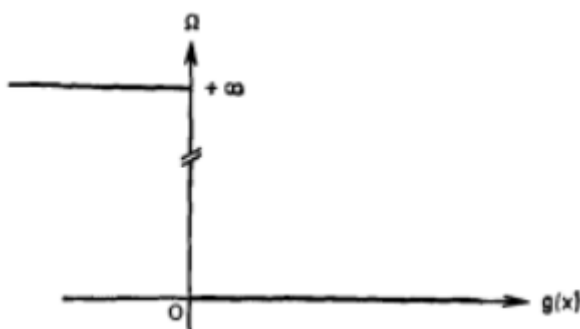


Рисунок 2.1 – Нескінченний бар'єр

б) логарифмічний штраф, який описується такою функцією:

$$Q(R, g(x)) = -R \ln g(x) \quad (2.12)$$

Це бар'єрна функція, яка є невизначеною в недопустимих точках (тобто коли $g(x) < 0$), тому потрібна спеціальна процедура, яка забезпечить попадання в допустиму для даної функції область;

в) обернений штраф, який показано на рисунку 2.2 і який описується такою функцією:

$$Q(R, g(x)) = \frac{R}{g(x)} \quad (2.13)$$

Даний штраф, як і попередній, є бар'єрною функцією;

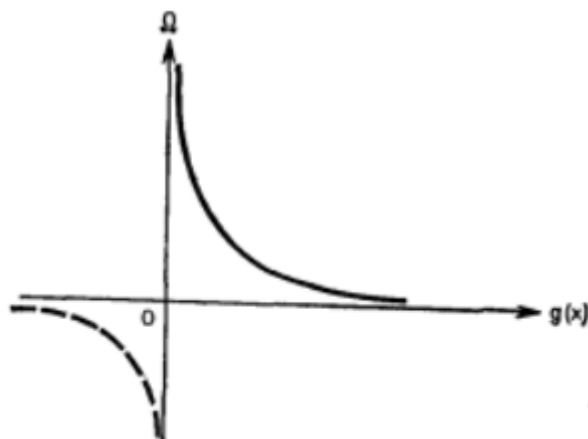


Рисунок 2.2 – Штраф, який задається оберненою функцією

г) штраф типу квадрату зрізання, який показано на рисунку 2.3 і який описується такою функцією:

$$Q(R, g(x)) = \begin{cases} 0, & g(x) \geq 0 \\ R(g(x))^2, & g(x) < 0 \end{cases} \quad (2.14)$$

Перш за все, слід зауважити, що даний штраф зовнішній, тому стаціонарні точки $P(x, R)$ можуть бути недопустимими. З іншої сторони, недопустимі точки не створюють в даному випадку додаткових складностей у порівнянні з допустимими: різниця між ними полягає лише в тому, що в допустимих і граничних точках штраф є рівним нулю. Саме з цих міркувань в програмному продукті було використано даний штраф в методі штрафних функцій.

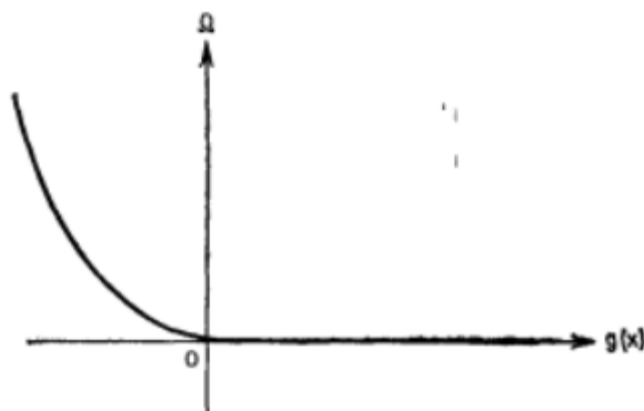


Рисунок 2.3 – Штраф типу квадрату зрізання

При використанні будь-якої з вище приведених функцій штрафу потрібно вибрати початкове значення параметра R і змінювати його після кожної підзадачі безумовної оптимізації задля того, щоб забезпечити збіжність послідовності стаціонарних точок x^t . Зрозуміло, що R потрібно змінювати так, щоб при переході з однією задачі безумовної оптимізації в іншу вагові коефіцієнти при функціях-обмеженнях, в яких точка увійшла в недопустиму область, збільшувались, а в інших зменшувались.

Таким чином, є доцільним використання такої процедури: почати оптимізацію з $R=0$ (безумовна оптимізація) і на кожній наступній підзадачі збільшувати R на певне число. Для програмного продукту даним числом було обрано 1, тобто на першій ітерації $R=0$, на другій $R=1$, на третій $R=2$ і так далі.

Побудова ефективних алгоритмів на основі методу штрафних функцій не представляє великої складності. Приведемо алгоритм, який було використано в програмному продукті:

а) задати початковий вектор x^0 , початковий параметр $R=0$, ε_1 – параметр закінчення одномірної оптимізації, ε_2 – параметр закінчення процедури безумовної оптимізації та ε_3 – параметр завершення роботи алгоритму;

б) побудувати штрафну функцію $P(x^t, R^t) = f(x^t) + Q(R^t, g(x^t))$;

в) знайти x^{t+1} , який є мінімумом функції $P(x, R)$ при фіксованому $R = R^t$. У якості початкової точки використовувати x^t , а в якості параметра закінчення кроку - ε_2 . В програмному продукті безумовна мінімізація відбувається методом спряжених градієнтів;

г) перевірити виконання умови $|P(x^{t+1}, R^t) - P(x^t, R^{t-1})| \leq \varepsilon_3$. У випадку, якщо вона виконується, то покласти $x^{t+1}=x^T$ і завершити процес розв'язання. Інакше, перейти до наступного кроку;

д) покласти $R^{t+1} = R^t + \Delta R$ у відповідності з обраним правилом перерахунку та перейти до пункту б.

2.4 Метод множників

Розглянемо функцію Лагранжа для ЗНП з обмеженнями у вигляді нерівностей:

$$L(x, u) = f(x) - \sum_{j=1}^m u_j g_j(x) \quad (2.15)$$

Проблема в тому, що, використовуючи функцію Лагранжа, не можна мінімізувати її по x та u з метою отримання (x^*, u^*) , оскільки вказаний вектор є не точкою мінімуму функції L , а її стаціонарною точкою. Виявляється, що якщо певним чином модифікувати функцію Лагранжа, то отримана в результаті функція буде досягати свого мінімуму в точці Куна-Такера вихідної задачі. Розглянемо метод модифікованих множників Лагранжа, який також називається методом множників [16, 20].

Розглянемо функцію

$$P(x, \sigma) = f(x) + R \sum_{j=1}^m (\langle g_j(x) + \sigma_j \rangle^2 - \sigma_j^2) \quad (2.16)$$

де R – постійний ваговий коефіцієнт;

$\langle \cdot \rangle$ - функція зрізання, яка має вид:

$$\langle x \rangle = \begin{cases} 0, & x \geq 0 \\ x, & x < 0 \end{cases} \quad (2.17)$$

Параметри σ_j виконують принципово іншу роль, ніж R , виконуючи зсув штрафних доданків. Даний зсув ітеративно уточнюється, в результаті чого процес збігається до розв'язку при не дуже жорстких обмеженнях на задачу. Початкова точка x^0 не обов'язково має бути допустимою, а початкове значення множників σ_j зручно брати 0.

Позначимо через x^t точку мінімуму штрафної функції, яка використовується на t -ій ітерації. При переході до $(t+1)$ -ої ітерації множники перераховуються за формулою:

$$\sigma_j^{t+1} = \langle g_j(x^t) + \sigma_j^t \rangle, j = 1, 2, \dots, m \quad (2.18)$$

Через наявність у формулі оператора зрізання вектор σ не має додатних компонент. Кожен новий вектор σ^t визначає зсув аргумента відповідного штрафу, причому формула перерахунку така, що в результаті зміни зсуву при переході до нової підзадачі штраф за порушення обмежень збільшується, і, як наслідок цього, стаціонарні точки x^t наближаються до допустимої області.

Для контролю збіжності метода використовують послідовності $x^t, \sigma^t, f(x^t)$ та $g(x^t)$. При цьому припускається, що алгоритм безумовної оптимізації, який використовується, кожний раз дозволяє знайти стаціонарну точку і після цього завершити ітерацію на основі певного критерію. Зупинка основного алгоритму відбувається тоді, коли хоча б одна з даних послідовностей перестає значно змінюватись при перерахунку множників і наступної безумовної оптимізації.

Побачимо, що градієнт штрафної функції в результаті безумовної оптимізації має в кінці стати нулем. Далі, за формулою (2.18) виконується перерахунок множників σ . Розглянемо величини σ^T і $g_j(x^T)$, вважаючи, що вони є границею відповідних послідовностей. За формулою (2.18) для існування граничних значень σ^T і $g_j(x^T)$ необхідно, щоб виконувались умови:

$$g_j(x^T) > 0 \text{ і } \sigma_j^T = 0, \text{ або } g_j(x^T) = 0 \text{ і } \sigma_j^T \leq 0 \quad (2.19)$$

Записавши градієнт штрафної функції, отримаємо:

$$\begin{cases} \nabla P(x^T) = \nabla f(x^T) + 2R \sum_{j=1}^m \sigma_j^T \nabla g_j(x^T) = 0 \\ g_j(x^T) \geq 0, j = 1, 2, \dots, m \\ \sigma_j^T g_j(x^T) = 0, j = 1, 2, \dots, m \\ \sigma_j^T \leq 0, j = 1, 2, \dots, m \end{cases} \quad (2.20)$$

Легко побачити, що задача (2.20) представляє собою умови Куна-Такера для точки x^T . Таким чином, отримана гранична точка є точкою Куна-Такера.

Таким чином, модифікований метод множників Лагранжа має такі основні властивості:

а) метод збігається до точки Куна-Такера. Проте, слід зауважити, що даний факт має місце, якщо можна вирішити підзадачі безумовної оптимізації;

б) при не дуже великих R задача мінімізації $P(x, R)$ по x має таку саму складність, як задача мінімізації функції $f(x)$;

в) правила перерахунку σ легкі і практично не потребують додаткових обчислень.

Проте даний метод має і свої недоліки. Один із них полягає в тому, що елементи ітераційної послідовності x^t наближаються до x^* , майже завжди знаходячись поза допустимою зоною. Також до недоліків слід віднести і відсутність правил вибору параметра R .

2.5 Метод комплексів

Розглянемо один із методів прямого пошуку, тобто пошук оптимальної точки буде проводитися лише обчисленням значень цільової функції та обмежень, не обчислюючи їх градієнти. Даний тип методів був розроблений у зв'язку з тим, що в прикладних технічних задачах цільова функція та обмеження можуть бути недиференційованими [5, 16].

Одним із яскравих прикладів методів даного типу є метод комплексів, який базується на прямому пошуку оптимальної точки вздовж симплексу. Було запропоновано будувати множину пробних P точок випадково і послідовно. Маючи попередньо задані інтервали для змінних x , можна на ЕОМ будувати їх випадкові реалізації, а саме, маючи верхні та нижні межі x_i^U і x_i^L , координати точок розраховуються за формулою:

$$x_i = x_i^L + r_i(x_i^U - x_i^L), i = 1, 2, \dots, n, \quad (2.21)$$

де r_i – випадкові числа, рівномірно розподілені на інтервалі $(0,1)$.

Кожна отримана точка перевіряється на допустимість, і якщо якийсь із обмежень порушується, то точка зміщується до центру мас вже побудованих точок до тих пір, поки дана точка не стане допустимою. Загальна кількість точок P має бути не менше, ніж $n+1$, але, зазвичай, вона є набагато більшою.

Після того як множина із P точок отримана, в кожній з них обчислюється цільова функція, і точка, яка має найбільше значення, відкидається. Нова точка отримується шляхом відображення відкидаємої точки через центр мас інших точок. Якщо x^R – точка, яку відкидаємо, а \bar{x} – центр мас точок, які залишилися, то нова точка визначається так:

$$x^m = \bar{x} + \alpha(\bar{x} - x^R) \quad (2.22)$$

Параметр α є відстанню відображення: при $\alpha=1$ має місце рівність $\|x^m - \bar{x}\| = \|\bar{x} - x^R\|$, $\alpha > 1$ відповідає розтягу, а $\alpha < 1$ – стиску.

Коли знайдена нова точка, значення цільової функції та обмежень, можливі наступні випадки:

а) нова точка допустима і значення цільової функції в ній не співпадає з максимальним значенням серед усієї множини точок. В такому випадку обираємо

точку з максимальним значенням функції та знову виконуємо операції відображення;

б) нова точка допустима і значення цільової функції в ній співпадає з максимальним значенням. Замість того, щоб виконувати відображення, змістимо дану точку на половину відстані до центру мас (якщо так не зробити, то можлива зациклованість);

в) нова точка недопустима. В цьому випадку зменшуємо вдвічі відстань до обчисленого раніше центру мас. Процедура пошуку продовжується до тих пір, поки багатогранник не буде стягнутий до центру мас в межах заданої точності чи поки різниця між значеннями цільової функції в вершинах не стане досить малою.

Були проведені числові експерименти по описаному алгоритму та на основі аналізу було виявлено, що краще обирати $\alpha > 1$ та $P > 2n$ [21]. Вибір α більшим за одиницю компенсує стиск комплексу, який спричинений зменшенням відстані до центру мас. Велике число вершин використовується для попередження виродження комплексу, коли пошук виконується біля границі допустимої області. Приведемо алгоритм, який було використано в програмному продукті.

Спочатку задана строго допустима початкова точка x^0 , параметр відображення α і параметри завершення обчислень ε та δ .

Крок 1. Побудова початкового комплексу, який складається з P допустимих точок. Для кожної точки $p=1,2,\dots,P$ виконуємо наступні операції:

а) випадковим чином генеруємо координати x_i^p ;

б) якщо x^p – недопустима точка, то треба знайти центр мас \bar{x} вже знайдених точок і покласти $x^p = x^p + \frac{1}{2}(\bar{x} - x^p)$, повторювати дану процедуру, поки x^p не стане допустимою;

в) якщо x^p – допустима точка, то повторювати пункт 1 до тих пір, поки $p=P$;

г) обчислити $f(x^p)$ для $p=1,\dots,P$.

Крок 2. Відображення комплексу:

а) обрати точку x^R , для якої $f(x^R) = \max_p f(x^p) = F_{max}$;

б) знайти центр мас \bar{x} і нову точку за формулою (2.22);

в) якщо x^m – недопустима точка і $f(x^m) \geq F_{max}$, зменшити відстань вдвічі між x^m і центром мас \bar{x} , і продовжувати дану операцію до тих пір, поки $f(x^m) < F_{max}$;

г) якщо x^m – допустима точка і $f(x^m) < F_{max}$, то перейти до кроку 4;

д) якщо x^m – недопустима точка, то перейти до кроку 3.

Крок 3. Якщо x^m – недопустима точка, то зменшити вдвічі відстань до центру мас і продовжувати до тих пір, поки вона не стане допустимою.

Крок 4. Перевірка умов завершення обчислень:

а) покласти $\bar{f} = \frac{1}{P} \sum_{p=1}^P f(x^p)$ і $\bar{x} = \frac{1}{P} \sum_{p=1}^P x^p$;

б) якщо $\|f(x^p) - \bar{f}\| \leq \varepsilon$ і $\|x^p - \bar{x}\| \leq \delta$, то завершити обчислення. Інакше, перейти до кроку 2.

Хоча метод комплексів не потребує диференційованості функцій, які визначають задачу, так як не потрібно шукати похідні від них, необхідною умовою є опуклість допустимої області. Дана умова є істотною при знаходженні центру мас і зміщенні допустимої, але не задовольняючої щодо значення цільової функції, точки. В першому випадку припускається, що центр мас - опукла комбінація допустимих точок – також має бути допустимим, як показано на рисунку 2.4. Аналогічно, при неопуклій області допустимих значень зміщена точка може стати недопустимою навіть, якщо центр мас допустимий, як показано на рисунку 2.5.

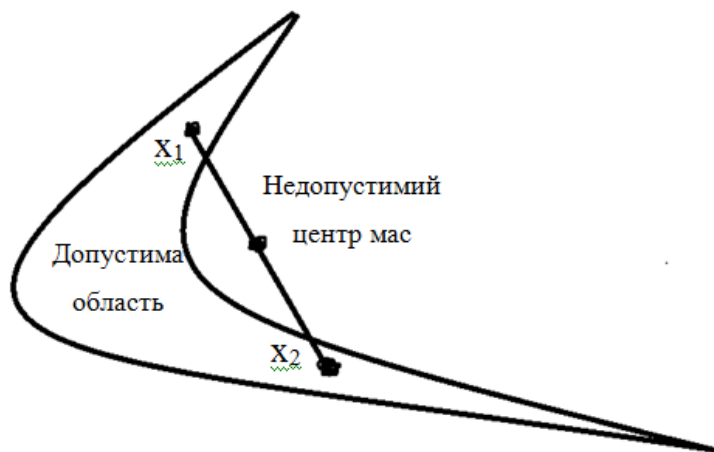


Рисунок 2.4 – Недопустимий центр мас



Рисунок 2.5 – Недопустима точка після зсуву

Таким чином, метод комплексів для неопуклих допустимих областей розбігається. На практиці метод широко використовується і дає гарні результати для опуклих задач, оскільки ситуації, описані вище, зустрічаються досить рідко.

2.6 Метод допустимих напрямків

Розглянемо ЗНП з обмеженнями у вигляді нерівностей:

$$\begin{cases} f(x) \rightarrow \min \\ g_j(x) \geq 0, j = 1, \dots, m \end{cases} \quad (2.23)$$

Нехай x^0 – початкова точка, яка задовольняє обмеженням, тобто $g_j(x^0) \geq 0, j = 1, \dots, m$. Припустимо, що деякі обмеження виконуються як рівності для x^0 . Нехай вектор d визначає підходящий напрям для пошуку, якщо d – напрямок спуску, тобто $\nabla f(x^0)d < 0$ і якщо точки променя $x(\alpha) = x^0 + \alpha d$, де $\alpha \geq 0$ є допустимими принаймні на невеликій відстані від x^0 . З точністю, яка забезпечується

лінійною апроксимацією, точки $x(\alpha)$ вздовж d будуть допустимими, якщо для всіх зв'язуючих в x^0 обмежень виконується умова

$$g_j(x^0) + \nabla g_j(x^0)(x - x^0) \geq 0 \quad (2.24)$$

Так як за припущенням $g_j(x^0) = 0$ і $x - x^0 = \alpha d$, де $\alpha \geq 0$, тоді остання умова еквівалентна умові для d :

$$\nabla g_j(x^0)d \geq 0 \quad (2.25)$$

для всіх $g_j(x)$, рівних нулю в точці x^0 . Напрямок d , який задовольняє записаним вище нерівностям, отримав назву допустимого напрямку [19, 22].

Основна ідея даного методу полягає в тому, щоби на кожному кроці алгоритму визначати вектор d як допустимий напрямок і скалярний параметр $\theta > 0$, що задовольняє умову:

$$\nabla f(x^t)d \leq -\theta \text{ і } \nabla g_j(x^t)d \geq \theta, \quad (2.26)$$

а значення θ обирається по можливості великим.

Після того як вектор напрямку обрано, наступне наближення може бути визначено пошуком по α вздовж прямої $x = x^t + \alpha d^t$ до тих пір, поки або $f(x)$ не досягне оптимуму, або яке-небудь з обмежень не стане порушеним. Зазвичай спочатку відбувається обчислення значення $\bar{\alpha}$, при якому яке-небудь з обмежень $g_j(x) \geq 0$ вперше виявляється зв'язуючим. При відомому $\bar{\alpha}$ можна використати будь-яку процедуру одномірного пошуку для визначення α , яке мінімізує функцію $f(x^t + \alpha d^t)$ на відрізку $[0, \bar{\alpha}]$.

Проте у даного алгоритму є один вагомий недолік. На практиці визначення множини активних обмежень, яка використовується в методі допустимих напрямків

$$I^t = \{j | 0 \leq g_j(x^t) \leq \varepsilon, j = 1, \dots, m\} \quad (2.27)$$

може не тільки уповільнити ітеративний процес, але і привести до збіжності до точок, які не є точками Куна-Такера. Дане явище обумовлено тим, що крок ітерації, який було отримано, стає все менше, а вектор напрямку коливається між близько розташованими границями. Саме тому було запропоновано модифікацію, яка називається методом Топкіса-Вейнотта [22]. Приведемо алгоритм даного методу, адже саме його було використано в програмному продукті.

Спочатку задана допустима початкова точка x^0 .

Крок 1. Розв'язати задачу лінійного програмування:

$$\begin{cases} \theta \rightarrow \max \\ \nabla f(x^t)d \leq -\theta \\ g_j(x^t) + \nabla g_j(x^t)d \geq 0, j = 1, 2, \dots, m \\ -1 \leq d_i \leq 1, i = 1, 2, \dots, n \end{cases} \quad (2.28)$$

Крок 2. Якщо $\theta^t \leq 0$, то обчислення завершити. Інакше, знайти $\bar{\alpha}$:

$$\bar{\alpha} = \min\{\alpha | g_j(x^t + \alpha d^t) = 0, j = 1, \dots, m \text{ і } \alpha \geq 0\} \quad (2.29)$$

Якщо не існує $\bar{\alpha} > 0$, то покласти $\bar{\alpha} = \infty$.

Крок 3. Знайти таке α^t , що

$$f(x^t + \alpha^t d^t) = \min\{f(x^t + \alpha d^t) | 0 \leq \alpha \leq \bar{\alpha}\}. \quad (2.30)$$

Покласти $x^{t+1} = x^t + \alpha^t d^t$ і продовжити розв'язок [23, 24].

Зауважимо, що основна відмінність між класичним методом допустимих напрямків та методом Топкіса-Вейнотта полягає у включенні $g_j(x^t)$ в нерівність в

підзадачі лінійного програмування. Якщо обмеження неактивно в x^t , тобто $g_j(x^t) > 0$, то j -е обмеження менше впливає на вибір d , оскільки додатне значення g_j зрівноважує вплив доданка, який містить градієнт. Така постановка підзадачі дозволяє уникнути неочікуваних змін в напрямку пошуку при наближенні до границі області допустимих значень.

Проте у випадку неопуклої допустимої області необхідно обережно проводити одномірний пошук, бо, рухаючись вздовж спочатку допустимого напрямку d , можна вийти за межі області допустимих значень, а потім знову повернутися до неї. Крім того, у випадку не опуклої цільової функції в ході одномірного пошуку можна зіткнутися із випадком багатоекстемальності. Зазвичай, береться перший зустрінутий локальний мінімум. За цих умов метод має таку властивість, що на кожній ітерації генеруються лише допустимі точки. Це дуже істотно для інженерних задач, в яких модель системи може бути невірною поза областю допустимих значень.

Висновки до розділу

У розділі розглянуто основні технічні характеристики, які мають бути присутні в моделі проектування електричного двигуна, досліджено цільову функцію системи та побудовано саму модель оптимального проектування однофазного асинхронного двигуна.

Також приведено базові факти з теорії нелінійного програмування, записано задачу Куна-Такера та пов'язані з нею теореми.

До того ж приведено декілька методів розв'язання задачі нелінійного програмування та проаналізовано їх переваги та недоліки.

Продемонстровано алгоритми, засновані на перетворенні задачі умовної оптимізації в безумовну, а саме, метод штрафних функцій та модифікований метод

множників Лагранжа. Розглянуто один з методів прямого пошуку - метод комплексів, а також досліджено метод допустимих напрямків та його модифікацію - метод Топкіса-Вейнотта.

РОЗДІЛ 3 АНАЛІЗ МЕТОДІВ НЕЛІНІЙНОГО ПРОГРАМУВАННЯ ТА ЇХ ПРАКТИЧНЕ ЗАСТОСУВАННЯ ПРИ ОПТИМАЛЬНОМУ ПРОЕКТУВАННІ ОДНОФАЗНОГО АСИНХРОННОГО ДВИГУНА ІЗ ЗАСТОСУВАННЯМ СТВОРЕНОГО ІНТЕРНЕТ-РЕСУРСУ

3.1 Порівняння методів нелінійного програмування

При моделюванні будь-якого процесу, перш за все необхідно дослідити якими методами можливо це виконати. Після цього потрібно порівняти ці методи, щоб розуміти який з них є швидким, який є точним, а іноді, потрібно навіть дослідити метод на робастність (можливість розв'язувати різні класи задач).

Таким чином для аналізу потрібних методів було використано декілька критеріїв, а саме: час роботи, допустимість отриманого результату та кількість ітерацій алгоритму.

Дане дослідження буде проводитись для двох задач нелінійного програмування.

Задача 1:

$$\begin{cases} (x_1 - 3)^2 + (x_2 - 3)^2 + (x_3 - 3)^2 \rightarrow \min \\ 100 - (x_1 - 3)^2 - (x_2 - 3)^2 - (x_3 - 3)^2 \geq 0 \\ x_1^3 + x_2^3 + x_3^3 \geq 0 \\ x_1 + x_2 + x_3 \geq 10 \end{cases} \quad (3.1)$$

Як видно одразу з умови, глобальним розв'язком безумовного варіанту цієї задачі є точка $x^* = (3; 3; 3)$. Проте ця точка не задовольняє третій умові.

В даній задачі усі змінні входять у вирази однаковим чином, тобто розв'язок буде мати рівні за значенням координати. Неважко помітити, що розв'язком буде точка, яка буде задовольняти третю нерівність зі знаком « $=$ ». Тобто розв'язком даної задачі є точка $x^* = (10/3; 10/3; 10/3)$.

Задача 2:

$$\begin{cases} (x_1 - 3)^2 + (x_2 - 3)^2 \rightarrow \min \\ 2x_1 - x_2^2 - 1 \geq 0 \\ 9 - 0.8x_1^2 - 2x_2 \geq 0 \end{cases} \quad (3.2)$$

Так як розмірність цієї задачі рівна двом, тоді зручно зобразити множину допустимих розв'язків, як показано на рисунку 3.1. Перетин двох фігур є даною множиною.

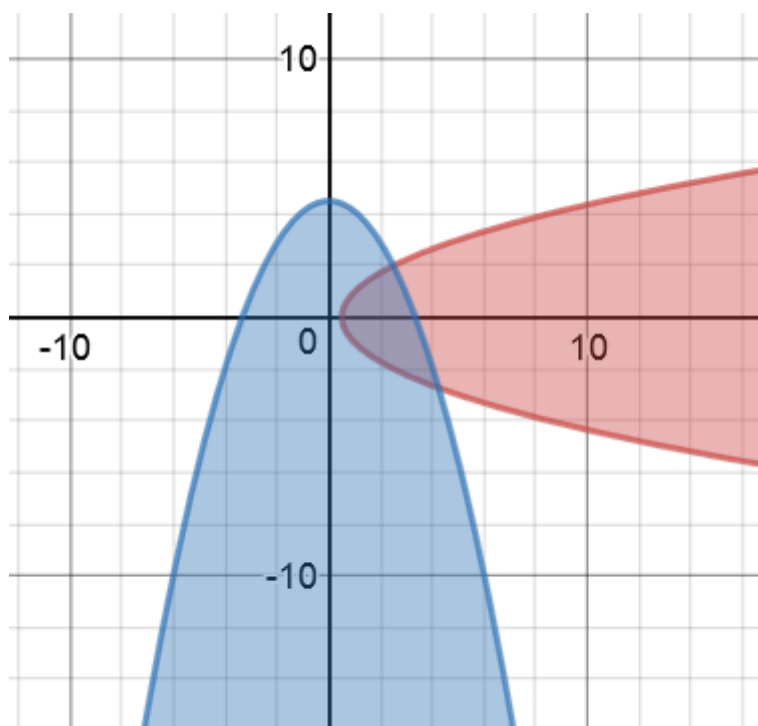


Рисунок 3.1 – Допустима множина точок другої задачі

Так як треба мінімізувати функцію

$$f = (x_1 - 3)^2 + (x_2 - 3)^2 \quad (3.3)$$

тобто потрібно знайти точку з допустимої множини, яка має найменшу відстань до точки (3;3). Як видно з рисунку 3.1 даною точкою є $x^* = (2.5; 2)$ - точка, яка має найбільше значення другої координати.

Тепер перейдемо до розв'язку даних задач обраними методами нелінійної оптимізації. Для порівняння методів було створено спеціальну консольну програму. Результати розв'язання будуть подаватися у такому порядку: початкова точка, значення цільової функції в початковій точці, значення всіх обмежень в початковій точці, кількість ітерацій алгоритму, оптимальна точка, значення цільової функції в оптимальній точці, значення всіх обмежень в оптимальній точці, час обчислень. Остаточні результати порівняння занесемо в таблицю 3.1, що знаходиться в кінці підрозділу.

Почнемо з методу штрафних функцій, результати якого зображені на рисунках 3.2 та 3.3:

```
First x is DenseVector 3-Double
1
1
1

Criterion is 12
Restriction 1 97
Restriction 2 3
Restriction 3 -7
12 iteration. Answer is DenseVector 3-Double
3,33223
3,33223
3,33223

Criterion is 0,331122174608345
Restriction 1) 66,6888113660869
Restriction 2) 111,000406005586
Restriction 3) -0,00332225678254616
RunTime 244,6007
```

Рисунок 3.2 – Розв'язок першої задачі методом штрафних функцій

```

First x is DenseVector 2-Double
1
1

Criterion is 8
Restriction 1 0
Restriction 2 6,2
10 iteration. Answer is DenseVector 2-Double
2,50031
2,00062

Criterion is 1,24843940753992
Restriction 1) -0,0018734857320144
Restriction 2) -0,00249795575739675
RunTime 158,4171

```

Рисунок 3.3 – Розв’язок другої задачі методом штрафних функцій

Тепер розв’яжемо задачі модифікованим методом множників Лагранжа, результати якого зображені на рисунках 3.4 та 3.5:

```

First x is DenseVector 3-Double
1
1
1

Criterion is 12
Restriction 1 97
Restriction 2 3
Restriction 3 -7
5 iteration. Answer is DenseVector 3-Double
3,33443
3,33443
3,33443

Criterion is 0,335537092249281
Restriction 1) 66,6446617513235
Restriction 2) 111,221153843941
Restriction 3) 0,00330019273786775
RunTime 226,8018

```

Рисунок 3.4 – Розв’язок першої задачі методом множників

```

First x is DenseVector 2-Double
1
1

Criterion is 8
Restriction 1 0
Restriction 2 6,2
5 iteration. Answer is DenseVector 2-Double
2,49975
1,9995

Criterion is 1,2512479632514
Restriction 1) 0,00149709058939074
Restriction 2) 0,00199607526120094
RunTime 177,3189

```

Рисунок 3.5 – Розв’язок другої задачі методом множників

Тепер розв’яжемо задачі методом комплексів. Даний метод є стохастичним, тому розв’язок кожен раз може змінюватись. Результати розв’язання даним методом зображені на рисунках 3.6 та 3.7:

```

First x is DenseVector 3-Double
4
4
4

Criterion is 3
Restriction 1 52
Restriction 2 192
Restriction 3 2
882 iteration. Answer is DenseVector 3-Double
3,33334
3,33327
3,3334

Criterion is 0,333333545822231
Restriction 1 66,6666646181026
Restriction 2 111,111121396334
Restriction 3 3,06012534423417E-07
RunTime 197,5754

```

Рисунок 3.6 – Розв’язок першої задачі методом комплексів

```

First x is DenseVector 2-Double
1
1

Criterion is 8
Restriction 1 0
Restriction 2 6,2
80 iteration. Answer is DenseVector 2-Double
2,50041
1,99669

Criterion is 1,25620967321752
Restriction 1 0,0140435625239133
Restriction 2 0,00495541354019213
RunTime 120,2096

```

Рисунок 3.7 – Розв’язок другої задачі методом комплексів

Після цього розв’яжемо задачі методом допустимих напрямків, результати якого зображені на рисунках 3.8 та 3.9:

```

First x is DenseVector 3-Double
4
4
4

Criterion is 3
Restriction 1 52
Restriction 2 192
Restriction 3 2
3307 iteration. Answer is DenseVector 3-Double
3,33474
3,33545
3,33475

Criterion is 0,336636828111187
Restriction 1) 66,6337080155373
Restriction 2) 111,275946768933
Restriction 3) 0,00494252605858847
RunTime 803,2459

```

Рисунок 3.8 – Розв’язок першої задачі методом допустимих напрямків


```

First x is DenseVector 2-Double
1
1

Criterion is 8
Restriction 1 0
Restriction 2 6,2
10 iteration. Answer is DenseVector 2-Double
2,49959
1,99877

Criterion is 1,25288208943272
Restriction 1) 0,00411259246934215
Restriction 2) 0,00411526810015639
RunTime 108,2701

```

Рисунок 3.9 – Розв’язок другої задачі методом допустимих напрямків

Таким чином, можна занести результати обчислень в таблиці 3.1 та 3.2:

Таблиця 3.1 – Результати обчислень першої задачі

Перша задача	Метод штрафних функцій	Метод множників	Метод комплексів	Метод допустимих напрямків
Час роботи алгоритму (мс)	244,6007	226,8081	197,5754	803,2459
Допустимість отриманого розв’язку	ні	так	так	так
Кількість ітерацій	12	5	882	3307

Таблиця 3.2 – Результати обчислень другої задачі

Друга задача	Метод штрафних функцій	Метод множників	Метод комплексів	Метод допустимих напрямків
Час роботи алгоритму (мс)	158,4171	177,3189	120,2096	108,2701
Допустимість отриманого розв’язку	ні	так	так	так

Продовження таблиці 3.2

Друга задача	Метод штрафних функцій	Метод множників	Метод комплексів	Метод допустимих напрямків
Кількість ітерацій	10	5	80	10

Після аналізу отриманих результатів, можна сказати, що метод штрафних функцій часто дає недопустимий, але близький до допустимої області розв'язок, бо, використовуючи штраф типу квадрата зрізання, цей метод стає так званим «методом зовнішньої точки» (коли точка, яка була отримана в кінці ітерації, може бути недопустимою). Метод допустимих напрямків містить велику кількість параметрів, які регулюють точність і які потрібно підбирати для кожної задачі окремо, через що він стає повільним та незручним.

Серед використаних методів на даних задачах краще за все продемонстрували себе метод множників та метод комплексів, причому найменшу кількість ітерацій дає метод множників, а час роботи алгоритму – метод комплексів.

3.2 Інтерфейс онлайн-ресурсу

Для оптимального проектування однофазного асинхронного двигуна було написано EngineOpt - прикладний Інтернет-ресурс на мові програмування C# із використанням платформи розробки веб-додатків ASP.NET (Active Server Pages for .NET) [25, 26]. Детально розглянемо інтерфейс та допоміжний функціонал створеного онлайн-ресурсу.

Створений сайт містить систему аутентифікації, тому при першому зверненні до нього користувач бачить сторінку входу в обліковий запис. На рисунку 3.10 можна побачити дану сторінку.

Рисунок 3.10 – Сторінка входу в обліковий запис

Дана платформа є багатомовною і підтримує три мови: українську, англійську та російську. На панелі керування сайту нагорі сайту можна побачити випадаючий список для вибору мову. На рисунку 3.11 можна побачити сторінку аутентифікації на іншій мові.

Рисунок 3.11 – Сторінка входу в обліковий запис англійською мовою

Якщо користувач ще не має своєї сторінки на даному сайті він може натиснути кнопку «Реєстрація» у правому верхньому кутку або натиснути на посилання «Реєстрація нового користувача» на сторінці аутентифікації. Дана сторінка зображена на рисунку 3.12.

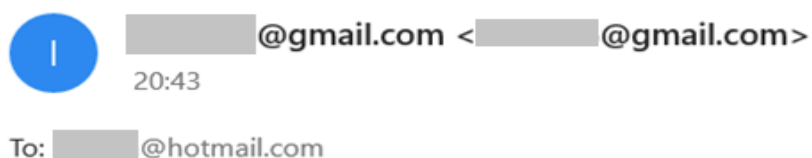
Рисунок 3.12 – Сторінка реєстрації нового облікового запису

Якщо реєстрація пройшла успішно, користувач побачить сторінку завершення реєстрації. Дана сторінка зображена на рисунку 3.13.

Рисунок 3.13 – Сторінка завершення реєстрації

Як видно з інформації, представленої на даній сторінці, на вказану електронну пошту буде відправлено лист із посиланням для завершення реєстрації на даному електронному ресурсі. На рисунку 3.14 можна побачити приклад листа, який буде відправлено після реєстрації для підтвердження електронної пошти користувача.

Підтвердження електронної пошти



Для завершення реєстрації перейдіть за посиланням: [завершити реєстрацію](#)

Рисунок 3.14 – Лист підтвердження електронної пошти

Після натискання посилання «завершити реєстрацію» в листі користувача буде переведено знову на даний сайт для входу в свою сторінку, де для цього йому

буде достатньо ввести свій логін та пароль. Якщо користувач забув свій пароль на сторінці аутентифікації (рис. 3.10) у нього є можливість змінити його. Для цього йому потрібно натиснути на посилання «Забули пароль?». Після цього відкриється сторінка нагадування паролю, на якій потрібно буде ввести електронну пошту свого облікового запису для відновлення паролю. Дана сторінка зображена на рисунку 3.15.

The screenshot shows the 'Забули пароль?' (Forgot Password) page. At the top is a dark red navigation bar with links: EngineOpt, Головне меню, Налаштування, Результати, Про сайт, and a language dropdown. On the right are links for 'Реєстрація' and 'Вхід'. Below the navigation bar, the title 'Забули пароль?' is followed by the instruction 'Введіть адресу вашої електронної пошти.' (Enter the address of your electronic mail). There is a text input field labeled 'Email' and a red button labeled 'Надіслати' (Send). At the bottom left, the footer text reads '26.09.2018 - EngineOpt'.

Рисунок 3.15 – Сторінка нагадування паролю

Якщо електронну пошту було введено вірно, користувач побачить сторінку відновлення паролю. Дану сторінку можна побачити на рисунку 3.16.

The screenshot shows the 'Відновлення паролю' (Password Reset) page. It features the same dark red navigation bar as the previous page. Below the navigation bar, the title 'Відновлення паролю' is followed by the instruction 'Будь ласка, перевірте вашу електронну пошту, щоб відновити пароль.' (Please check your electronic mail to reset the password). At the bottom left, the footer text reads '26.09.2018 - EngineOpt'.

Рисунок 3.16 – Сторінка відновлювання паролю

Як видно з інформації, представленої на даній сторінці, на вказану електронну пошту буде відправлено лист із посиланням для відновлення паролю свого облікового запису. На рисунку 3.17 можна побачити приклад листа, який буде відправлено після введення своєї електронної пошти для відновлення паролю.

Скидання паролю



To: [redacted]@hotmail.com

Для скидання паролю аккаунта з логіном "Qwerty123" перейдіть за посиланням: [скинути пароль](#)

Рисунок 3.17 – Лист скидання паролю

Після натискання посилання «скинути пароль» в листі користувача буде переведено знову на даний сайт для створення нового паролю для свого облікового запису. Сторінку скидання паролю можна побачити на рисунку 3.18.

Рисунок 3.18 – Сторінка скидання паролю

Після заповнення форми та натискання кнопки «Скидання» користувача буде переведено на сторінку аутентифікації, а його пароль в базі буде змінено на той, що він ввів на даній сторінці.

Тепер розглянемо дії користувача, коли він увійшов у свій обліковий запис. При натисканні на кнопку «Про сайт» на панелі керування сайту буде відображено сторінку інформації про роботу електронного ресурсу EngineOpt. На цій сторінці описано функціонал даного сайту та описано алгоритм роботи із ним. Дана сторінка зображена на рисунку 3.19.

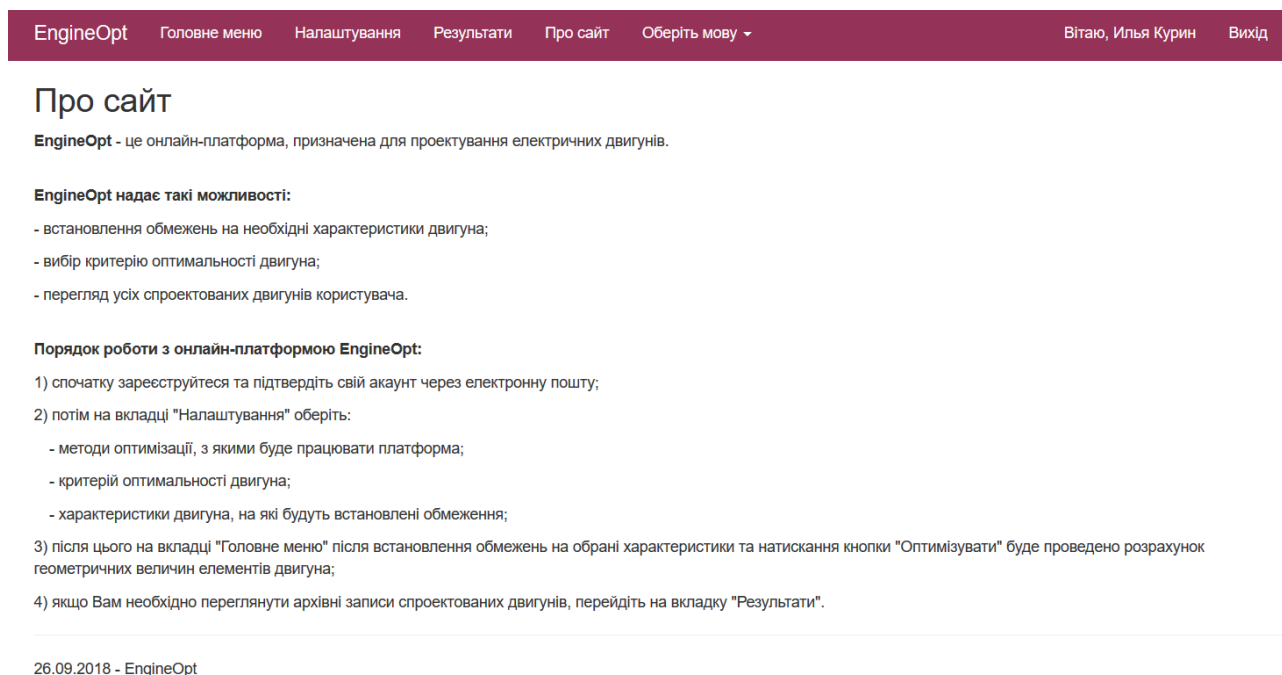


Рисунок 3.19 – Сторінка інформації про сайт

При натисканні на кнопку «Налаштування» на панелі керування сайту буде відображено сторінку налаштувань моделі майбутньої задачі оптимізації. Дану сторінку можна побачити на рисунку 3.20.

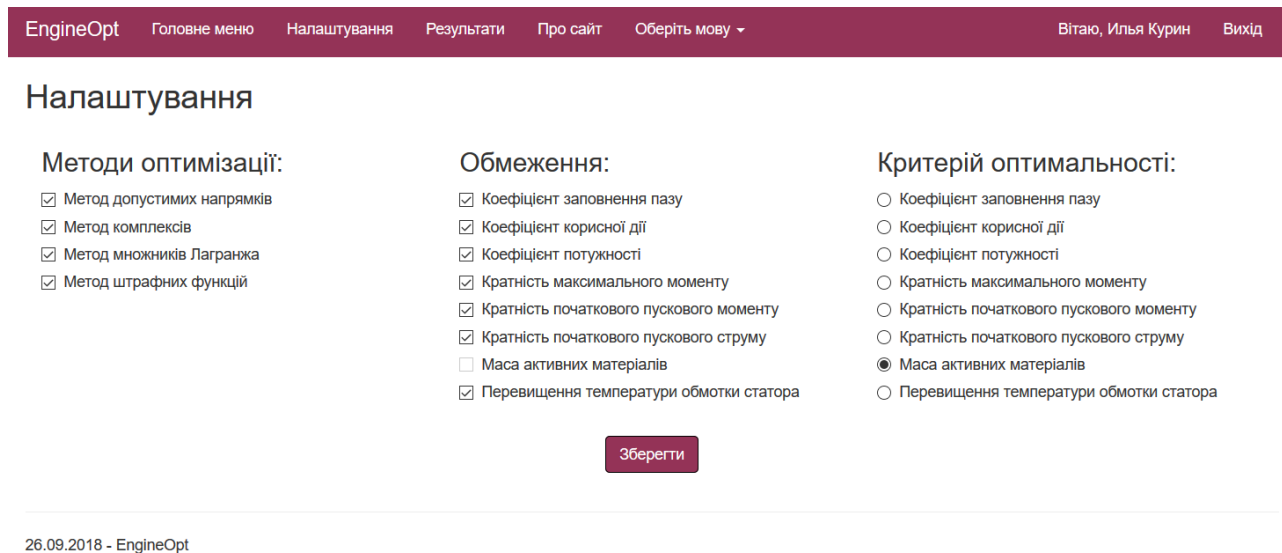


Рисунок 3.20 – Сторінка налаштувань сайту

При натисканні на кнопку «Головне меню» на панелі керування сайту буде відображено сторінку для створення оптимального електричного двигуна, де будуть вводиться потрібні обмеження на його технічні та технологічні характеристики. Дана сторінка зображена на рисунку 3.21.

EngineOpt

Головне меню

Налаштування

Результати

Про сайт

Оберіть мову

Вітаю, Ілья Курин

Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

Kmax ≥	1,6
Kп ≥	0,5
Ki ≤	4
Kзап ≥	0,7
KCosφi ≥	0,9
ККД ≥	0,58
Тоб ≤	75

bzr =	
bzs =	
Da =	
Ds =	
har =	
has =	
Ls =	

Оптимізувати

Kmax - Кратність максимального моменту, Kп - Кратність початкового пускового моменту, Ki - Кратність початкового пускового струму, Kзап - Коефіцієнт заповнення пазу, KCosφi - Коефіцієнт потужності, ККД - Коефіцієнт корисної дії, Тоб - Перевищення температури обмотки статора

bzr - Ширина зубця ротора, bzs - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

26.09.2018 - EngineOpt

Рисунок 3.21 – Сторінка створення двигуна

При натисканні на кнопку «Результати» на панелі керування сайту буде відображено сторінку з таблицею, на якій зображено технічні характеристики усіх створених електричних двигунів даним користувачем. Цю сторінку можна побачити на рисунку 3.21.

EngineOpt

Головне меню

Налаштування

Результати

Про сайт

Оберіть мову

Вітаю, Ілья Курин

Вихід

Результати

Двигуни									
Дата	Час	Діаметр розточення статора	Довжина пакета статора	Ширина зубця статора	Ширина зубця ротора	Висота стінки статора	Висота стінки статора	Зовнішній діаме	
04.08.2018	21:47	0.07561058717122766	0.03899923620611557	0.004410201598762557	0.0078715663437104	0.06360795776910469	0.023793993099293365	0.110246036406	
04.08.2018	22:04	0.06175557767940032	0.05113104149522896	0.0029540375738140376	0.0051971422270663935	0.07048607691737765	0.01866098322387205	0.106027328030	
04.08.2018	22:06	0.056141433591354706	0.05516647513712384	0.002908954150024654	0.004045245188476678	0.06531167787778988	0.022346579198687377	0.102135431101	
04.08.2018	22:06	0.059339849341024514	0.058670229087729425	0.0026522947649491218	0.00465486138700474	0.05668787159995564	0.015919810027028365	0.092485968342	
04.08.2018	22:11	0.06647710123401	0.050875539943052245	0.0035334248508016694	0.00586164762375953	0.06628184859467756	0.015582332561529397	0.118729131346	
04.08.2018	22:12	0.05753814411822273	0.07075028542627254	0.0032446596754172373	0.0037167023019316547	0.05236594082753581	0.012026508965426166	0.094684258263	
04.08.2018	22:27	0.07177309187989227	0.04357036623057358	0.003829229827867743	0.007089693455983885	0.07300220285728115	0.023679231723789242	0.117859630469	
06.08.2018	20:36	0.07178446607757344	0.06834106209925418	0.003819379729500584	0.004152217481678093	0.08156325582055311	0.015718117261797778	0.153931216719	
09.08.2018	20:53	0.0727273676917508	0.05999637739931977	0.0034367657663666802	0.0069571417823098905	0.05813217143017068	0.017369834485722952	0.088325159186	
09.08.2018	20:53	0.06792843950779877	0.07497825639945285	0.004007869021848255	0.00548966820660658	0.05139771426551839	0.014184197611768503	0.097972310997	

ДивитисьПошукОновити

Стр. 1 з 12

26.09.2018 - EngineOpt

Рисунок 3.22 – Сторінка результатів проектування двигунів

3.3 Робота з онлайн-ресурсом

Для демонстрації можливостей онлайн-ресурсу EngineOpt відкриємо сторінку «Про сайт», що зображена на рисунку 3.19. Розглянемо список «Порядок роботи з онлайн-платформою EngineOpt». Спочатку користувач має створити свій обліковий запис та увійти у нього. Дії для виконання цієї задачі було описано в попередньому пункті розділу. Тепер перейдемо безпосередньо до роботи із сайтом. Як написано у другому пункті списку треба перейти до вкладки «Налаштування» і обрати хоча б один із запропонованих методів нелінійної оптимізації для розв’язання майбутньої задачі, хоча б одне обмеження на задачу, тобто характеристики двигуна та один критерій якості. Якщо користувач не обрав жодного методу оптимізації і натиснув кнопку «Зберегти», платформа попереджає, що потрібно обрати хоча б один. Таке повідомлення зображено на рисунку 3.23.

Рисунок 3.23 – Попередження про відсутність обраних методів оптимізації

Якщо користувач обрав хоча б один метод оптимізації, то під заголовком сторінки він отримає повідомлення про успішне збереження даних. Це повідомлення показано на рисунку 3.24.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Налаштування

Налаштування були збережені.

Методи оптимізації:

- ☒ Метод допустимих напрямків
- ☐ Метод комплексів
- ☐ Метод множників Лагранжа
- ☐ Метод штрафних функцій

Обмеження:

- ☒ Коефіцієнт заповнення пазу
- ☒ Коефіцієнт корисної дії
- ☒ Коефіцієнт потужності
- ☒ Кратність максимального моменту
- ☒ Кратність початкового пускового моменту
- ☒ Кратність початкового пускового струму
- ☐ Маса активних матеріалів
- ☒ Перевищення температури обмотки статора

Критерій оптимальності:

- ☐ Коефіцієнт заповнення пазу
- ☐ Коефіцієнт корисної дії
- ☐ Коефіцієнт потужності
- ☐ Кратність максимального моменту
- ☐ Кратність початкового пускового моменту
- ☐ Кратність початкового пускового струму
- ☒ Маса активних матеріалів
- ☐ Перевищення температури обмотки статора

Зберегти

27.09.2018 - EngineOpt

Рисунок 3.24 – Повідомлення про успішне збереження налаштувань

Аналогічним чином побудована логіка збереження налаштувань для обмежень задачі та критерію оптимальності, тобто користувач має обрати хоча б одну характеристику із другого стовпця та єдину характеристику із третього стовпця на даній сторінці.

Тепер, користуючись порядком дій на сторінці «Про сайт», перейдемо до вкладки «Головне меню», що зображена на рисунку 3.21. На даній сторінці в лівій таблиці ми можемо записувати обмеження на ті характеристики, які обрали на вкладці «Налаштування». Внизу сторінки записані пояснення до скорочень обраних технічних характеристик та геометричних параметрів, які необхідно знайти. На рисунку 3.25 зображено сторінку «Головне меню», коли користувач обрав не усі можливі обмеження. Також, числа, які введе користувач мають задовольняти обмеження, задані в програмному коді, зроблені для того, щоб задача мала фізичний сенс. Дані обмеження можна побачити на рисунку 3.26.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберть мову
Вітаю, Ілля Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	$b_{zr} =$	
$K_{\text{зап}} \geq$	0,7	$b_{zs} =$	
$T_{\text{об}} \leq$	75	$D_a =$	
		$D_s =$	
		$h_{ar} =$	
		$h_{as} =$	
		$L_s =$	

Оптимізувати

K_{\max} - Кратність максимального моменту, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, $T_{\text{об}}$ - Перевищення температури обмотки статора,
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

27.09.2018 - EngineOpt

Рисунок 3.25 – Сторінка «Головне меню» із неповним списком обмежень

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберть мову
Вітаю, Ілля Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1 - 2	$b_{zr} =$	
$K_{\text{п}} \geq$	0,2 - 0,7	$b_{zs} =$	
$K_i \leq$	2 - 5	$D_a =$	
$K_{\text{зап}} \geq$	0,5 - 0,9	$D_s =$	
$K_{\text{Cos}\phi} \geq$	0,8 - 0,95	$h_{ar} =$	
$K_{\text{КД}} \geq$	0,5 - 0,9	$h_{as} =$	
$T_{\text{об}} \leq$	50 - 100	$L_s =$	

Оптимізувати

K_{\max} - Кратність максимального моменту, $K_{\text{п}}$ - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, $K_{\text{Cos}\phi}$ - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

27.09.2018 - EngineOpt

Рисунок 3.26 – Обмеження на запис в таблиці головного меню

Коли користувач ввів усі обмеження на характеристики, йому необхідно натиснути на кнопку «Оптимізувати», щоб програма почала розв'язувати задачу нелінійної оптимізації тими методами, які обрав користувач на сторінці «Налаштування». Після закінчення розрахунків платформа надасть результати в таблицю «Оптимальні значення», де записано значення отриманих геометричних параметрів двигуна, та поле «Результати», де записано значення обраного критерію

оптимальності та характеристик. Результат розв'язання поставленої задачі зображено на рисунку 3.27.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	bzr =	0,0056831515	Отриманий двигун має такі параметри: Маса активних матеріалів = 4,94901908530784 Кратність максимального моменту = 1,75299322142533 Кратність початкового пускового моменту = 1,08873915101433 Кратність початкового пускового струму = 3,97785914376674 Коефіцієнт заповнення пазу = 1,80529898842879 Коефіцієнт потужності = 0,999997117139435 Коефіцієнт корисної дії = 0,630790867393171 Перевищення температури обмотки статора = 11,0932447213166 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу допустимих напрямків.
$K_p \geq$	0,5	bzs =	0,0026501319	
$K_i \leq$	4	Da =	0,1353257259	
$K_{\text{зап}} \geq$	0,7	Ds =	0,065373396	
$K_{\text{Cosfi}} \geq$	0,9	har =	0,0007868423	
$K_{\text{КД}} \geq$	0,58	has =	0,0828325083	
$T_{\text{об}} \leq$	75	Ls =	0,059481802	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
bzs - Ширина зубця ротора, bzs - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.27 – Результати проектування електричного двигуна

Після цього, користуючись порядком дій на сторінці «Про сайт», перейдемо до вкладки «Результати», що зображена на рисунку 3.22. На ній побачимо велику таблицю, стовпцями якої є значення геометричних параметрів та характеристик електричних двигунів, створених користувачем через поточний обліковий запис. До того ж, першими двома колонками є дата і час створеного двигуна, щоб користувачеві було в майбутньому легше його шукати. Також на цій вкладці ми можемо обирати кількість двигунів, зображуваних на одній сторінці, та переміщуватися між сторінками за допомогою кнопок внизу вкладки. Знайдемо попередньо створений двигун в даній таблиці та натиснемо на нього мишкою. Це зображено на рисунку 3.28.

EngineOpt

Головне меню

Налаштування

Результати

Про сайт

Оберіть мову

Вітаю, Ілья Курин

Вихід

Результати

Двигуни

Дата	Час	Діаметр розточения статора	Довжина пакета статора	Ширина зубця статора	Ширина зубця ротора	Висота стінки статора	Висота стінки статора	Зовнішній діам
15.10.2018	13:24	0.07355428113158904	0.04490921186202712	0.004356025349343963	0.007620850020584602	0.06552408541893892	0.017098184118342134	0.117514529593
15.10.2018	13:25	0.0609389540742689	0.044828227911637375	0.0038839726101844376	0.00574163631919883	0.06745919401580679	0.014222960272917516	0.1111892668799
15.10.2018	13:25	0.06581813451106172	0.056670089462766714	0.002759750424153281	0.006191071620293241	0.06384017356352899	0.014077303635965344	0.108151085065
15.10.2018	13:25	0.06432429452628098	0.044360766445454565	0.003163495125325162	0.006111190727265174	0.06877283082286494	0.016431089935093693	0.114388760693
15.10.2018	13:26	0.07022524127747176	0.0673291162947794	0.0038575887833058784	0.006109959202171284	0.07017636181329208	0.021039354493394193	0.121680338127
15.10.2018	13:26	0.07176437172859949	0.03686493971181374	0.0044838707660062	0.007062425130938622	0.0849024204387011	0.022705113492304575	0.159616102186
15.10.2018	13:30	0.06867540601416247	0.06005466421356461	0.003927148792688171	0.006055124721037209	0.05705927394041094	0.015394933804960827	0.099096047562
15.10.2018	22:46	0.068953132172559	0.0652447820525825	0.003264198588330391	0.005622614085731382	0.09990772763262863	0.01589678744501285	0.166920370122
15.10.2018	22:46	0.06537339597398122	0.05948180199994785	0.0026501318668234	0.005683151534397024	0.08283250829682495	0.0007868422954218745	0.135325725941

ДивитисьПошукНовити

Стор2932910

кнопку «Дивитись» в лівому нижньому кутку таблиці. Отримане вікно зображене на рисунку 3.30.

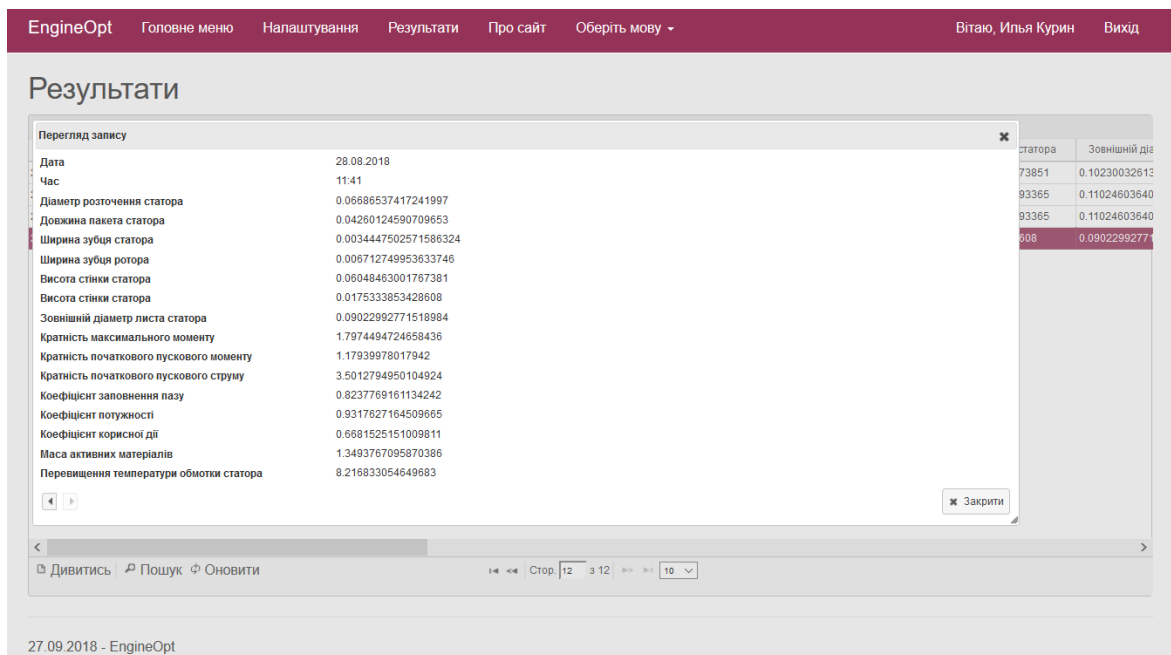


Рисунок 3.30 – Перегляд запису таблиці результатів

Також в даній таблиці є можливою фільтрація даних за обраним критерієм. Для цього необхідно натиснути кнопку «Пошук» внизу таблиці. Відкрите вікно зображене на рисунку 3.31.

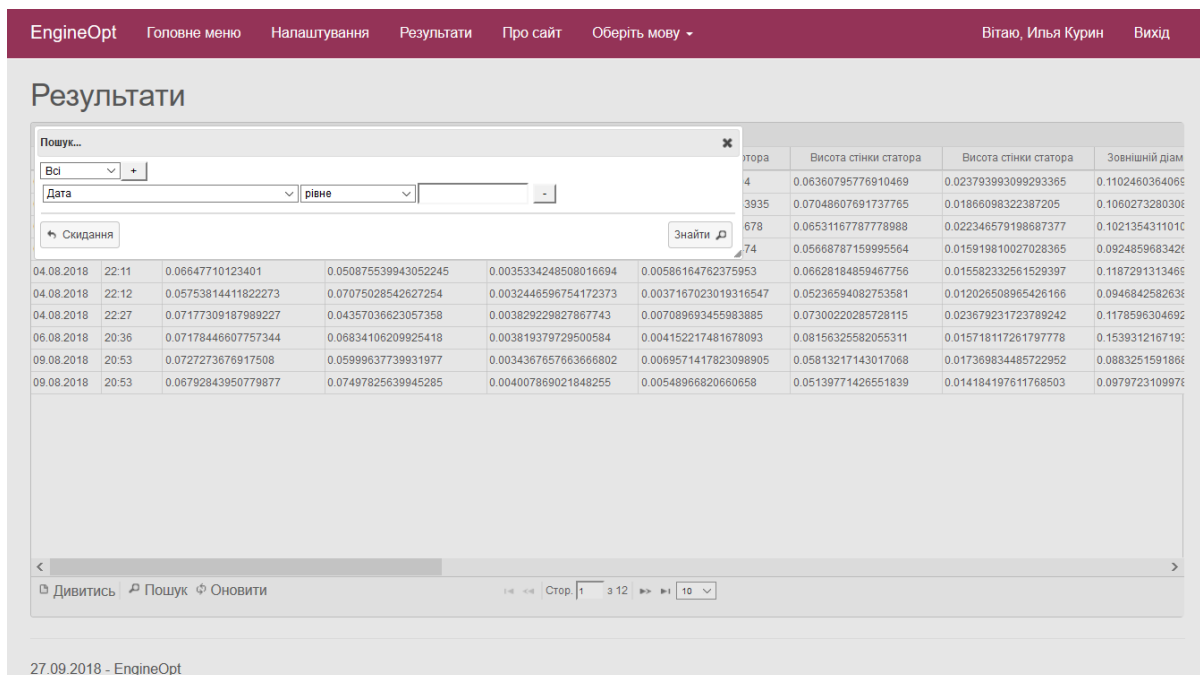


Рисунок 3.31 – Вікно пошуку запису в таблиці результатів

Профільтруємо дані таким чином, що будуть відображені лише записи двигунів, які спроектовані 6 серпня 2018 року. Результат фільтрації зображений на рисунку 3.32.

The screenshot shows the 'Результати' (Results) page of the EngineOpt website. At the top, there is a dark red navigation bar with links: EngineOpt, Головне меню, Налаштування, Результати, Про сайт, and a language selector. The user is logged in as 'Вітаю, Ілля Курин' with a 'Вихід' (Logout) link.

The main section is titled 'Результати' and contains a table of engine specifications. The table has the following columns: Дата, Час, Діаметр розточення статора, Довжина пакета статора, Ширина зубця статора, Ширина зубця ротора, Висота стінки статора, Висота стінки ротора, and Зовнішній діаметр. The first row of data shows values for 06.08.2018 at 20:36.

Below the table, there is a search filter window titled 'Пошук...'. It contains a dropdown menu set to 'Всі', a date field set to '06.08.2018', and a 'Знайти' (Find) button. There is also a 'Скидання' (Reset) button.

At the bottom of the page, there is a footer with the date '27.09.2018 - EngineOpt'.

Рисунок 3.32 – Фільтрація по даті

Тепер для того, щоб відобразити знову усі дані необхідно закрити вікно пошуку та натиснути кнопку «Оновити» внизу сторінки.

3.4 Пошук допустимої точки

Тепер перейдемо безпосередньо до розв’язання задач нелінійної оптимізації за допомогою онлайн-ресурсу EngineOpt.

Для задач умовної оптимізації використовують три методи пошуку початкової допустимої точки: випадковий пошук, мінімізація штрафної функції при відсутності обмежень та послідовна мінімізація при наявності обмежень. На даному сайті було використано метод випадкового пошуку. Він полягає в генеруванні пробних точок із перевіркою на їхню допустимість.

По заданим верхнім та нижнім межам кожної координати шуканого вектору, які попередньо відомі з певних фізичних міркувань, обчислюються

$$x_i = x_i^- + r_i(x_i^+ - x_i^-), i = 1, \dots, n, \quad (3.4)$$

де r_i - випадкові числа, рівномірно розподілені на інтервалі $[0; 1]$;

x_i^+ і x_i^- - верхні та нижні межі відповідно кожної координати.

Отримана точка перевіряється на допустимість обчисленням функцій, які є обмеженнями в даній задачі.

Проте постає проблема, коли допустима множина є пустою. Вона може призвести до за циклювання програми. Тому в програмному продукті одна процедура пошуку початкової точки включає 1000000 випадкових пошуків. Як тільки визначається допустима точка, то процедура закінчується. Інакше, програма обирає допустиму точку серед оптимальних точок вже до цього створених двигунів через поточний обліковий запис, за умови, що такі точки будуть задовольняти поточну систему обмежень. Якщо ж платформа знайшла множину точок, які задовольняють дані умови, допустимою точкою стає та, яка має найкраще значення поточного критерію оптимальності, у порівнянні із іншими точками знайденої множини. Подібне явище можна побачити на рисунку 3.33, де на таблиці з результатами проектування п'ятий, шостий і сьомий запис створених двигунів співпадають саме через те, що була знайдена однакова початкова допустима точка.

EngineOpt

Головне меню

Налаштування

Результати

Про сайт

Оберіть мову

Вітаю, Ілья Курин

Вихід

Результати

Двигуни								
Дата	Час	Діаметр розточення статора	Довжина пакета статора	Ширина зубця статора	Ширина зубця ротора	Висота стінки статора	Висота стінки статора	Зовнішній діаме
13.08.2018	19:57	0.08045920215399399	0.04445719110727364	0.004906573802471817	0.009089096897721262	0.0687324889709791	0.031306258438064616	0.115425715794
13.08.2018	19:57	0.06593377394227537	0.054843070993592535	0.003231253377592729	0.006070425728601602	0.09497517441983155	0.02274518091266285	0.122560548034
13.08.2018	19:58	0.0820588960202156	0.03434635136350617	0.005126496708267389	0.009534535886366272	0.08407358317585595	0.01782238707561135	0.146717938827
13.08.2018	20:56	0.06452084783711623	0.043549481056347	0.004409840192798817	0.006076068489417945	0.056034834742755016	0.006085865508116957	0.091597830480
16.08.2018	21:02	0.07561058717122766	0.03899923620611557	0.004410201598762557	0.0078715663437104	0.06360795776910469	0.023793993099293365	0.110246036406
16.08.2018	21:03	0.07561058717122766	0.03899923620611557	0.004410201598762557	0.0078715663437104	0.06360795776910469	0.023793993099293365	0.110246036406
16.08.2018	21:05	0.07561058717122766	0.03899923620611557	0.004410201598762557	0.0078715663437104	0.06360795776910469	0.023793993099293365	0.110246036406
16.08.2018	21:06	0.07750586586676977	0.03395895001152761	0.005289247939661831	0.009006427661422083	0.0591884452944643	0.004908853120130979	0.1023321111141
16.08.2018	21:11	0.06678145597193991	0.0660034880700715	0.00356392307529887	0.005769653816753631	0.05675458911210881	0.013830782148092104	0.095791818569
16.08.2018	21:15	0.07561058717122766	0.03899923620611557	0.004410201598762557	0.0078715663437104	0.06360795776910469	0.023793993099293365	0.110246036406

<

Дивитись

Пошук

Оновити

16

3 42

10

04.12.2018 - EngineOpt

Рисунок 3.33 – Збіг початкових допустимих точок

Якщо знайдена множина точок є пустою, то користувач отримає повідомлення, яке зображено на рисунку 3.34.

EngineOpt

Головне меню

Налаштування

Результати

Про сайт

Оберіть мову

Вітаю, Ілья Курин

Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

Kmax ≥	2
Kп ≥	0,7
Kі ≤	2
Kзап ≥	0,9
KCosfi ≥	0,95
ККД ≥	0,9
Тоб ≤	50

bzr =	
bzs =	
Da =	
Ds =	
har =	
has =	
Ls =	

Допустима множина, ймовірно, пуста.

Оптимізувати

Kmax - Кратність максимального моменту, Kп - Кратність початкового пускового моменту, Kі - Кратність початкового пускового струму, Kзап - Коефіцієнт заповнення пазу, KCosfi - Коефіцієнт потужності, ККД - Коефіцієнт корисної дії, Тоб - Перевищення температури обмотки статора

bzr - Ширина зубця ротора, bzs - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

29.09.2018 - EngineOpt

Рисунок 3.34 – Пуста допустима множина

3.5 Оптимальне проектування однофазного асинхронного двигуна з використанням онлайн-ресурсу

Тепер, коли порівняно всі запропоновані методи нелінійної оптимізації на прикладі тривіальної задачі та знайдено спосіб пошуку початкової допустимої точки, можна приступити безпосередньо до оптимального проектування. Для цього буде проведено серію проектувань двигунів із різними допустимими множинами, утвореними від технічних обмежень двигуна, та різними критеріями оптимальності, використовуючи різні методи нелінійної оптимізації.

Спочатку, проведемо проектування двигуна із критерієм оптимальності – мінімізацією маси активних матеріалів та такими допустимими величинами обмежень:

- кратність максимального моменту ($K_{\max} \geq 1.6$);
- кратність початкового пускового моменту ($K_{\Pi} \geq 0.5$);
- кратність початкового пускового струму ($K_I \leq 4$);
- коефіцієнт заповнення пазу ($K_{\text{зап}} \geq 0.7$);
- коефіцієнт потужності ($\text{COS}\phi_i \geq 0.9$);
- коефіцієнт корисної дії ($\text{ККД} \geq 0.58$);
- перевищення температури обмоток статора ($T_{\text{об}} \leq 75$).

Послідовно створимо двигун з такими обмеженнями кожним з приведених методів окремо. Результати приведені на рисунках 3.35-3.38.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	$b_{zr} =$	0,0053212207	Отриманий двигун має такі параметри: Маса активних матеріалів = 2,24561240226501 Кратність максимального моменту = 1,71181378552737 Кратність початкового пускового моменту = 1,03751098063697 Кратність початкового пускового струму = 3,85055026162688 Коефіцієнт заповнення пазу = 1,81405927318008 Коефіцієнт потужності = 0,999999982127172 Коефіцієнт корисної дії = 0,642123502663287 Перевищення температури обмотки статора = 5,78566543052511 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу допустимих напрямків.
$K_{\phi} \geq$	0,5	$b_{zs} =$	0,0046241882	
$K_i \leq$	4	$D_a =$	0,0945576136	
$K_{\text{зап}} \geq$	0,7	$D_s =$	0,0639651263	
$K_{\text{Cosfi}} \geq$	0,9	$h_{ar} =$	0,0109732643	
$K_{\text{КД}} \geq$	0,58	$h_{as} =$	0,0485160802	
$T_{\text{об}} \leq$	75	$L_s =$	0,0614545947	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{ϕ} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.35 – Проектування першого двигуна за допомогою методу допустимих напрямків

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	$b_{zr} =$	0,0068346207	Отриманий двигун має такі параметри: Маса активних матеріалів = 1,59261844554598 Кратність максимального моменту = 1,60107038825474 Кратність початкового пускового моменту = 1,02079108397283 Кратність початкового пускового струму = 3,38500063751063 Коефіцієнт заповнення пазу = 0,723102374812527 Коефіцієнт потужності = 0,900148088523275 Коефіцієнт корисної дії = 0,681802816933164 Перевищення температури обмотки статора = 15,314119744564 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу комплексів.
$K_{\phi} \geq$	0,5	$b_{zs} =$	0,0038499148	
$K_i \leq$	4	$D_a =$	0,0989145202	
$K_{\text{зап}} \geq$	0,7	$D_s =$	0,0688174004	
$K_{\text{Cosfi}} \geq$	0,9	$h_{ar} =$	0,0207495557	
$K_{\text{КД}} \geq$	0,58	$h_{as} =$	0,0640914779	
$T_{\text{об}} \leq$	75	$L_s =$	0,037312252	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{ϕ} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.36 – Проектування першого двигуна за допомогою методу комплексів

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0072522202$	Отриманий двигун має такі параметри: Маса активних матеріалів = 5,30004017549959 Кратність максимального моменту = 1,7853803223925 Кратність початкового пускового моменту = 1,11273451736178 Кратність початкового пускового струму = 3,67873617468981 Коефіцієнт заповнення пазу = 0,9026692402165 Коефіцієнт потужності = 0,939145346412621 Коефіцієнт корисної дії = 0,60316553349658 Перевищення температури обмотки статора = 54,1847539996585 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу множників Лагранжа.
$K_p \geq 0,5$	$b_{zs} = 0,0040487786$	
$K_i \leq 4$	$Da = 0,1621038956$	
$K_{\text{зап}} \geq 0,7$	$Ds = 0,0722426829$	
$K_{\text{Cosfi}} \geq 0,9$	$har = 0,0200337633$	
$K_{\text{КД}} \geq 0,58$	$has = 0,0936148991$	
$T_{\text{об}} \leq 75$	$Ls = 0,0384217475$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.37 – Проектування першого двигуна за допомогою методу множників

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0072794217$	Отриманий двигун має такі параметри: Маса активних матеріалів = 4,80557516035714 Кратність максимального моменту = 1,61800115197122 Кратність початкового пускового моменту = 0,984987696074384 Кратність початкового пускового струму = 3,90294345226788 Коефіцієнт заповнення пазу = 0,709736747431176 Коефіцієнт потужності = 0,96978170508846 Коефіцієнт корисної дії = 0,650979290507335 Перевищення температури обмотки статора = 28,3716076230412 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу штрафних функцій.
$K_p \geq 0,5$	$b_{zs} = 0,0044351447$	
$K_i \leq 4$	$Da = 0,1566185026$	
$K_{\text{зап}} \geq 0,7$	$Ds = 0,0740737923$	
$K_{\text{Cosfi}} \geq 0,9$	$har = 0,0152355463$	
$K_{\text{КД}} \geq 0,58$	$has = 0,0911375797$	
$T_{\text{об}} \leq 75$	$Ls = 0,0409257581$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.38 – Проектування першого двигуна за допомогою методу штрафних функцій

Як видно з результатів роботи програми, проектування двигуна із потрібними технологічними обмеженнями було зроблено усіма методами, а найкращі результати дали методи допустимих напрямків та комплексів.

Тепер проведемо проектування двигуна із такими наступними величинами обмежень:

- кратність максимального моменту (K_{\max}) ≥ 1.8 ;
- кратність початкового пускового моменту (K_{Π}) ≥ 0.5 ;
- кратність початкового пускового струму (K_I) ≤ 4 ;
- коефіцієнт заповнення пазу ($K_{\text{зап}}$) ≥ 0.7 ;
- коефіцієнт потужності (K_{COSfi}) ≥ 0.9 ;
- коефіцієнт корисної дії ($K_{\text{КД}}$) ≥ 0.6 ;
- перевищення температури обмоток статора ($T_{\text{об}}$) ≤ 70 .

Послідовно створимо двигун з такими обмеженнями кожним з приведених методів окремо. Результати приведені на рисунках 3.39-3.42.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,8$	$b_{zr} = 0,0072380042$	Отриманий двигун має такі параметри: Маса активних матеріалів = 4,47321070878413 Кратність максимального моменту = 1,81335728974496 Кратність початкового пускового моменту = 1,17386545243402 Кратність початкового пускового струму = 3,9463149193926 Коефіцієнт заповнення пазу = 0,740812404920705 Коефіцієнт потужності = 0,987217081381084 Коефіцієнт корисної дії = 0,640673922611778 Перевищення температури обмотки статора = 4,91507267936842 Допустима точка була знайдена за допомогою випадкового пошуку. Результат не був отриманий жодним із методів.
$K_{\Pi} \geq 0,5$	$b_{zs} = 0,0034473647$	
$K_I \leq 4$	$D_a = 0,1369725908$	
$K_{\text{зап}} \geq 0,7$	$D_s = 0,0715546086$	
$K_{\text{Cosfi}} \geq 0,9$	$h_{ar} = 0,0223982171$	
$K_{\text{КД}} \geq 0,6$	$h_{as} = 0,0863711789$	
$T_{\text{об}} \leq 70$	$L_s = 0,0466330311$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{Π} - Кратність початкового пускового моменту, K_I - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора
 b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.39 – Проектування другого двигуна за допомогою методу допустимих напрямків

Головне меню

Обмеження

$K_{max} \geq$	1,8
$K_{\Pi} \geq$	0,5
$K_i \leq$	4
$K_{зап} \geq$	0,7
$K_{Cosfi} \geq$	0,9
$ККД \geq$	0,6
$Тоб \leq$	70

Оптимальні значення

$b_{zr} =$	0,0090529003
$b_{zs} =$	0,0053825274
$Da =$	0,1023405277
$Ds =$	0,0777038544
$har =$	0,0050145455
$has =$	0,0593807639
$Ls =$	0,0340755561

Результати

Отриманий двигун має такі параметри:
 Маса активних матеріалів = 0,777726197418603
 Кратність максимального моменту = 2,03820854203849
 Кратність початкового пускового моменту = 1,35012776000018
 Кратність початкового пускового струму = 3,23956559331823
 Коефіцієнт заповнення пазу = 0,701246158348081
 Коефіцієнт потужності = 0,909614480116088
 Коефіцієнт корисної дії = 0,600515928404641
 Перевищення температури обмотки статора = 44,0587881007
 Допустима точка була отримана з бази.
 Результат було отримано за допомогою методу комплексів.

Оптимізувати

K_{max} - Кратність максимального моменту, K_{Π} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{зап}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $ККД$ - Коефіцієнт корисної дії, $Тоб$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.40 – Проектування другого двигуна за допомогою методу комплексів

Головне меню

Обмеження

$K_{max} \geq$	1,8
$K_{\Pi} \geq$	0,5
$K_i \leq$	4
$K_{зап} \geq$	0,7
$K_{Cosfi} \geq$	0,9
$ККД \geq$	0,6
$Тоб \leq$	70

Оптимальні значення

$b_{zr} =$	0,0057952662
$b_{zs} =$	0,0033754203
$Da =$	0,0978350661
$Ds =$	0,0618826236
$har =$	0,0168099602
$has =$	0,0642555928
$Ls =$	0,0472564974

Результати

Отриманий двигун має такі параметри:
 Маса активних матеріалів = 1,91969062022124
 Кратність максимального моменту = 1,88519575121092
 Кратність початкового пускового моменту = 1,24500422257556
 Кратність початкового пускового струму = 3,42658884890274
 Коефіцієнт заповнення пазу = 0,729473827276887
 Коефіцієнт потужності = 0,932442327249575
 Коефіцієнт корисної дії = 0,641872679608781
 Перевищення температури обмотки статора = 23,8972652451647
 Допустима точка була знайдена за допомогою випадкового пошуку.
 Результат було отримано за допомогою методу множників Лагранжа.

Оптимізувати

K_{max} - Кратність максимального моменту, K_{Π} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{зап}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $ККД$ - Коефіцієнт корисної дії, $Тоб$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.41 – Проектування другого двигуна за допомогою методу множників

Головне меню

Обмеження

$K_{\max} \geq$	1,8
$K_{\Pi} \geq$	0,5
$K_I \leq$	4
$K_{\text{зап}} \geq$	0,7
$K_{\text{Cosfi}} \geq$	0,9
$K_{\text{КД}} \geq$	0,6
$T_{\text{об}} \leq$	70

Оптимальні значення

$b_{zr} =$	0,0070327414
$b_{zs} =$	0,0037303509
$D_a =$	0,0947945523
$D_s =$	0,070358319
$h_{ar} =$	0,0173528354
$h_{as} =$	0,0520420436
$L_s =$	0,0538085488

Результати

Отриманий двигун має такі параметри:
 Маса активних матеріалів = 2,43673830839572
 Кратність максимального моменту = 1,89194692462565
 Кратність початкового пускового моменту = 1,19867164645981
 Кратність початкового пускового струму = 3,86605927834874
 Коефіцієнт заповнення пазу = 90,7605067139917
 Коефіцієнт потужності = 0,999999975349747
 Коефіцієнт корисної дії = 0,601369268115536
 Перевищення температури обмотки статора = 0,915705950931282
 Допустима точка була знайдена за допомогою випадкового пошуку.
 Результат не був отриманий жодним із методів.

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{Π} - Кратність початкового пускового моменту, K_I - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, $T_{\text{об}}$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

15.10.2018 - EngineOpt

Рисунок 3.42 – Проектування другого двигуна за допомогою методу штрафних функцій

Як бачимо, методи допустимих напрямків та штрафних функцій не дали результатів. Це пояснюється тим, що в функціях, які описують обмеження, відбулося ділення на число, близьке до нуля, через що обчислення вивели алгоритм за межі допустимої області для методів штрафних функцій та допустимих напрямків

Тепер, проведемо проектування двигуна із критерієм оптимальності – максимізацією коефіцієнту корисної дії та такими допустимими величинами обмежень:

- кратність максимального моменту (K_{\max}) ≥ 1.6 ;
- кратність початкового пускового моменту (K_{Π}) ≥ 0.5 ;
- кратність початкового пускового струму (K_I) ≤ 4 ;
- коефіцієнт заповнення пазу ($K_{\text{зап}}$) ≥ 0.7 ;
- коефіцієнт потужності (COSfi) ≥ 0.9 ;
- маса активних матеріалів (M) ≤ 4 ;
- перевищення температури обмоток статора ($T_{\text{об}}$) ≤ 75 .

Послідовно створимо двигун з такими обмеженнями кожним з приведених методів окремо. Результати приведені на рисунках 3.43-3.46.

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0062874874$	<p>Отриманий двигун має такі параметри:</p> <p>Коефіцієнт корисної дії = 0,532598407609048</p> <p>Кратність максимального моменту = 2,17123502896364</p> <p>Кратність початкового пускового моменту = 1,43184553791716</p> <p>Кратність початкового пускового струму = 3,65485641099999</p> <p>Коефіцієнт заповнення пазу = 1,57566894710337</p> <p>Коефіцієнт потужності = 0,999992783795071</p> <p>Маса активних матеріалів = 0,519379550801248</p> <p>Перевищення температури обмотки статора = 3,08625286303887</p> <p>Допустима точка була знайдена за допомогою випадкового пошуку.</p> <p>Результат було отримано за допомогою методу допустимих напрямків.</p>
$K_p \geq 0,5$	$b_{zs} = 0,0044833023$	
$K_i \leq 4$	$D_a = 0,0753227075$	
$K_{\text{зап}} \geq 0,7$	$D_s = 0,0648535961$	
$K_{\text{Cosfi}} \geq 0,9$	$h_{ar} = 0,0005572674$	
$M \leq 4$	$h_{as} = 0,0409451865$	
$T_{\text{об}} \leq 75$	$L_s = 0,0597654046$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, M - Маса активних матеріалів, $T_{\text{об}}$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.43 – Проектування третього двигуна за допомогою методу допустимих напрямків

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0042424987$	<p>Отриманий двигун має такі параметри:</p> <p>Коефіцієнт корисної дії = 0,687353314146726</p> <p>Кратність максимального моменту = 1,60393285160553</p> <p>Кратність початкового пускового моменту = 0,956366687207039</p> <p>Кратність початкового пускового струму = 3,71107159840339</p> <p>Коефіцієнт заповнення пазу = 3,11181994616794</p> <p>Коефіцієнт потужності = 0,967030926106463</p> <p>Маса активних матеріалів = 3,40310447744157</p> <p>Перевищення температури обмотки статора = 9,77939933453198</p> <p>Допустима точка була знайдена за допомогою випадкового пошуку.</p> <p>Результат було отримано за допомогою методу комплексів.</p>
$K_p \geq 0,5$	$b_{zs} = 0,0035289036$	
$K_i \leq 4$	$D_a = 0,0984801349$	
$K_{\text{зап}} \geq 0,7$	$D_s = 0,057857831$	
$K_{\text{Cosfi}} \geq 0,9$	$h_{ar} = 0,023396551$	
$M \leq 4$	$h_{as} = 0,0530036481$	
$T_{\text{об}} \leq 75$	$L_s = 0,0545749864$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, M - Маса активних матеріалів, $T_{\text{об}}$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.44 – Проектування третього двигуна за допомогою методу комплексів

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0056459923$	<p>Отриманий двигун має такі параметри:</p> <p>Коефіцієнт корисної дії = 0,428032595341048</p> <p>Кратність максимального моменту = 2,93593375507391</p> <p>Кратність початкового пускового моменту = 1,97494614002741</p> <p>Кратність початкового пускового струму = 2,88060023779849</p> <p>Коефіцієнт заповнення пазу = 1,7978854491459</p> <p>Коефіцієнт потужності = 0,935888048265321</p> <p>Маса активних матеріалів = 2,78688391564904</p> <p>Перевищення температури обмотки статора = 7,97363968318736</p> <p>Допустима точка була знайдена за допомогою випадкового пошуку.</p> <p>Результат було отримано за допомогою методу множників Лагранжа.</p>
$K_{\phi} \geq 0,5$	$b_{zs} = 0,0038557515$	
$K_i \leq 4$	$Da = 0,0988986888$	
$K_{\text{зап}} \geq 0,7$	$Ds = 0,0568196101$	
$K_{\text{Cosfi}} \geq 0,9$	$har = 0,0157716947$	
$M \leq 4$	$has = 0,05324543$	
$T_{\text{об}} \leq 75$	$Ls = 0,0548182272$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{ϕ} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, M - Маса активних матеріалів, $T_{\text{об}}$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.45 – Проектування третього двигуна за допомогою методу множників

EngineOpt Головне меню Налаштування Результати Про сайт Оберіть мову Вітаю, Ілья Курин Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,0060245341$	<p>Отриманий двигун має такі параметри:</p> <p>Коефіцієнт корисної дії = 0,63477711307799</p> <p>Кратність максимального моменту = 1,60247345631837</p> <p>Кратність початкового пускового моменту = 0,964249514311595</p> <p>Кратність початкового пускового струму = 3,36824185048776</p> <p>Коефіцієнт заповнення пазу = 1,70309944325362</p> <p>Коефіцієнт потужності = 0,901772475955498</p> <p>Маса активних матеріалів = 3,50700140697372</p> <p>Перевищення температури обмотки статора = 8,28140599366614</p> <p>Допустима точка була знайдена за допомогою випадкового пошуку.</p> <p>Результат не був отриманий жодним із методів.</p>
$K_{\phi} \geq 0,5$	$b_{zs} = 0,0041867039$	
$K_i \leq 4$	$Da = 0,1312902583$	
$K_{\text{зап}} \geq 0,7$	$Ds = 0,0659918831$	
$K_{\text{Cosfi}} \geq 0,9$	$har = 0,0183608738$	
$M \leq 4$	$has = 0,0708180993$	
$T_{\text{об}} \leq 75$	$Ls = 0,0381486343$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{ϕ} - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, M - Маса активних матеріалів, $T_{\text{об}}$ - Перевищення температури обмотки статора

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.46 – Проектування третього двигуна за допомогою методу штрафних функцій

Згідно з результатами роботи програми, проектування двигуна із потрібними технологічними обмеженнями було зроблено усіма методами, окрім методу штрафних функцій.

Також проведемо проектування двигуна із критерієм оптимальності – мінімізацією перевищення температури обмотки статора та такими допустимими величинами обмежень:

- кратність максимального моменту (K_{\max}) ≥ 1.6 ;
- кратність початкового пускового моменту (K_{Π}) ≥ 0.5 ;
- кратність початкового пускового струму (K_I) ≤ 4 ;
- коефіцієнт заповнення пазу ($K_{\text{зап}}$) ≥ 0.7 ;
- коефіцієнт потужності ($\text{COS}\phi$) ≥ 0.9 ;
- коефіцієнт корисної дії (ККД) ≥ 0.58 ;
- маса активних матеріалів (M) ≤ 4 .

Послідовно створимо двигун з такими обмеженнями кожним з приведених методів окремо. Результати приведені на рисунках 3.47-3.50.

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову ▾
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження	Оптимальні значення	Результати
$K_{\max} \geq 1,6$	$b_{zr} = 0,005158115$	<p>Отриманий двигун має такі параметри:</p> <p>Перевищення температури обмотки статора = 82,9209856404773</p> <p>Кратність максимального моменту = 1,77073434657912</p> <p>Кратність початкового пускового моменту = 1,11225326868115</p> <p>Кратність початкового пускового струму = 3,84202463942905</p> <p>Коефіцієнт заповнення пазу = 0,828617408836892</p> <p>Коефіцієнт потужності = 0,995752662466416</p> <p>Коефіцієнт корисної дії = 0,608792593466132</p> <p>Маса активних матеріалів = 3,64290434578704</p> <p>Допустима точка була знайдена за допомогою випадкового пошуку.</p> <p>Результат було отримано за допомогою методу допустимих напрямків.</p>
$K_{\Pi} \geq 0,5$	$b_{zs} = 0,003503885$	
$K_I \leq 4$	$D_a = 0,1070167775$	
$K_{\text{зап}} \geq 0,7$	$D_s = 0,0632051783$	
$\text{KCos}\phi \geq 0,9$	$h_{ar} = 0,0175564323$	
$\text{ККД} \geq 0,58$	$h_{as} = 0,0672611672$	
$M \leq 4$	$L_s = 0,0685079314$	

Оптимізувати

K_{\max} - Кратність максимального моменту, K_{Π} - Кратність початкового пускового моменту, K_I - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, $\text{KCos}\phi$ - Коефіцієнт потужності, ККД - Коефіцієнт корисної дії, M - Маса активних матеріалів

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.47 – Проектування четвертого двигуна за допомогою методу допустимих напрямків

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	$b_{zr} =$	0,0060601019	Отриманий двигун має такі параметри: Перевищення температури обмотки статора = 51,6277821723682 Кратність максимального моменту = 1,81578501722551 Кратність початкового пускового моменту = 1,17965336814851 Кратність початкового пускового струму = 3,86639791837971 Коефіцієнт заповнення пазу = 0,711634591694417 Коефіцієнт потужності = 0,99723230839437 Коефіцієнт корисної дії = 0,643872473880564 Маса активних матеріалів = 2,07942275697027 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу комплексів.
$K_p \geq$	0,5	$b_{zs} =$	0,0038963924	
$K_i \leq$	4	$D_a =$	0,0965207589	
$K_{\text{зап}} \geq$	0,7	$D_s =$	0,0655401294	
$K_{\text{Cosfi}} \geq$	0,9	$h_{ar} =$	0,0172801455	
$K_{\text{КД}} \geq$	0,58	$h_{as} =$	0,0617584172	
$M \leq$	4	$L_s =$	0,0560180718	

Оптимізувати

16.10.2018 - EngineOpt

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, M - Маса активних матеріалів

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

Рисунок 3.48 – Проектування четвертого двигуна за допомогою методу комплексів

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

$K_{\max} \geq$	1,6	$b_{zr} =$	0,0063035551	Отриманий двигун має такі параметри: Перевищення температури обмотки статора = 62,3117096829865 Кратність максимального моменту = 1,62630617107552 Кратність початкового пускового моменту = 1,01050295385349 Кратність початкового пускового струму = 3,92861662918706 Коефіцієнт заповнення пазу = 0,700451278690783 Коефіцієнт потужності = 0,993829219342923 Коефіцієнт корисної дії = 0,682321387868592 Маса активних матеріалів = 2,843383713551 Допустима точка була знайдена за допомогою випадкового пошуку. Результат було отримано за допомогою методу множників Лагранжа.
$K_p \geq$	0,5	$b_{zs} =$	0,0047313327	
$K_i \leq$	4	$D_a =$	0,112745148	
$K_{\text{зап}} \geq$	0,7	$D_s =$	0,0686538264	
$K_{\text{Cosfi}} \geq$	0,9	$h_{ar} =$	0,0180622678	
$K_{\text{КД}} \geq$	0,58	$h_{as} =$	0,065911477	
$M \leq$	4	$L_s =$	0,0511827613	

Оптимізувати

16.10.2018 - EngineOpt

K_{\max} - Кратність максимального моменту, K_p - Кратність початкового пускового моменту, K_i - Кратність початкового пускового струму, $K_{\text{зап}}$ - Коефіцієнт заповнення пазу, K_{Cosfi} - Коефіцієнт потужності, $K_{\text{КД}}$ - Коефіцієнт корисної дії, M - Маса активних матеріалів

b_{zr} - Ширина зубця ротора, b_{zs} - Ширина зубця статора, D_a - Зовнішній діаметр листа статора, D_s - Діаметр розточення статора, h_{ar} - Висота стінки ротора, h_{as} - Висота стінки статора, L_s - Довжина пакета статора

Рисунок 3.49 – Проектування четвертого двигуна за допомогою методу множників

EngineOpt
Головне меню
Налаштування
Результати
Про сайт
Оберіть мову
Вітаю, Ілья Курин
Вихід

Головне меню

Обмеження

Оптимальні значення

Результати

Kmax ≥	1,6	bzr =	0,006689426	Отриманий двигун має такі параметри: Перевищення температури обмотки статора = 64,4832027364195 Кратність максимального моменту = 1,64291603741753 Кратність початкового пускового моменту = 1,03213330048767 Кратність початкового пускового струму = 3,93887147070275 Коефіцієнт заповнення пазу = 0,700420406519612 Коефіцієнт потужності = 0,997779848332368 Коефіцієнт корисної дії = 0,667941149338238 Маса активних матеріалів = 0,948904377884403 Допустима точка була знайдена за допомогою випадкового пошуку. Результат не був отриманий жодним із методів.
Kп ≥	0,5	bzs =	0,0042109892	
Kі ≤	4	Da =	0,0775314947	
Kзап ≥	0,7	Ds =	0,0708150628	
KCosfi ≥	0,9	har =	0,0188896386	
KКД ≥	0,58	has =	0,0520915514	
M ≤	4	Ls =	0,0580533788	

Оптимізувати

Kmax - Кратність максимального моменту, Kп - Кратність початкового пускового моменту, Kі - Кратність початкового пускового струму, Kзап - Коефіцієнт заповнення пазу, KCosfi - Коефіцієнт потужності, KКД - Коефіцієнт корисної дії, M - Маса активних матеріалів

bzr - Ширина зубця ротора, bzs - Ширина зубця статора, Da - Зовнішній діаметр листа статора, Ds - Діаметр розточення статора, har - Висота стінки ротора, has - Висота стінки статора, Ls - Довжина пакета статора

16.10.2018 - EngineOpt

Рисунок 3.50 – Проектування четвертого двигуна за допомогою методу штрафних функцій

Згідно з результатами роботи програми, проектування двигуна із потрібними технологічними обмеженнями було зроблено усіма методами, окрім методу штрафних функцій.

Таким чином, оптимально спроектувавши усі чотири двигуни, кожен з яких мав свої технічні та технологічні обмеження, а також, використовуючи різні критерії оптимальності в деяких двигунах, можна сказати, що методи штрафних функцій та допустимих напрямків є досить нестійкими щодо обчислень, бо в процесі ділення у функціях обмежень на числа, які близькі до нуля, відбувається вихід мінімізуючої послідовності алгоритму з меж допустимої області. Гарно себе показав метод комплексів, за умови початку алгоритму з точки, яка знаходиться не дуже близько до границь допустимої області. Проте найкращим можна вважати модифікований метод множників Лагранжа, адже він дав гарні результати з кожної початкової точки для даних систем обмежень та критеріїв оптимальності.

Висновки до розділу

В даному розділі було порівняно методи нелінійної оптимізації на прикладі двох задач умовної оптимізації з використанням консольної версії програмного продукту.

Також було продемонстровано інтерфейс та функціонал створеного Інтернет-ресурсу для оптимального проектування однофазного асинхронного двигуна.

Після цього було запропоновано метод пошуку початкової допустимої точки для прикладних технічних задач.

В кінці розділу було показано результати проектування онлайн-платформою EngineOpt за допомогою різних методів оптимізації та зроблено висновок щодо якості розв'язків приведеними алгоритмами.

РОЗДІЛ 4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ

4.1 Опис ідеї проекту

Спочатку проаналізуємо та подамо у вигляді таблиці зміст ідеї стартап-проекту, можливі напрямки застосування та основні вигоди, які може отримати користувач товару. Ці характеристики стартап-проекту зображено в таблиці 4.1.

Таблиця 4.1 - Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Програмна онлайн-платформа для пошуку оптимальних геометричних розмірів складових частин електричних двигунів машин.	1. Створення двигунів на підприємствах важкої промисловості.	Можливість користувачеві обирати потрібні характеристики двигуна, що надасть йому змогу витратити менше часу на проектування, збільшити потужність та зменшити витрати на виробництво двигунів.
	2. Створення двигунів на підприємствах з машинобудівництва.	Можливість витратити менше коштів на виробництво двигунів та позиціонування на екологічності власної продукції.

Тепер зробимо аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів. Результати аналізу зображено в таблиці 4.2.

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко- економічні характери- тики ідеї	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конку- рент 1	Конку- рент 2			
1.	Ціна	2000\$/ рік	4000\$/ рік	5000\$/ рік			+
2.	Прибутки	30000\$/ рік	40000\$/ рік	20000\$/ рік		+	
3.	Контроль якості	Аналі- тики та прог- рамісти	Аналі- тики, прог- рамісти та деякі клієнти	Прог- рамісти			+
4.	Динаміка галузі	Швид- ка	Пові- льна	Швид- ка		+	
5.	Постійні витрати	10000\$/ рік	20000\$/ рік	15000\$/ рік			+
6.	Змінні витрати	5000\$ - 10000\$/ рік	1000\$ - 2000\$/ рік	2000\$ - 5000\$/ рік	+		
7.	Патенти на продукти	Немає	Патент на кож- ний проект	Декі- лька патен- тів на винахід	+		
8.	Гнучкі ціни	Ціна єдина	Ціна варію- ється з року в рік	Ціна єдина		+	

Продовження таблиці 4.2

№ п/п	Техніко- економічні характери- стики ідеї	Товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конку- рент 1	Конку- рент 2			
9.	Законо- давчі обмеження	Немає	Немає	Обме- ження на кіль- кість розро- бників			+

4.2 Технологічний аудит ідеї проекту

Визначимо технологічну здійсненність ідеї проекту за допомогою аналізу таких складових, як технології, за якою буде виготовлено товар згідно ідеї проекту, існування таких технологій, чи їх необхідно розробити / доробити, доступність таких технологій авторам проекту. Результати даного аналізу зображено в таблиці 4.3.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
Програмна онлайн-платформа для пошуку оптимальних геометричних розмірів складових частин електричних двигунів машин, які використовуються в побуті.	Технологія створення інтернет-додатків ASP.NET із використанням нейронних мереж.	Так.	Дані технології доступні, проте необхідно багато часу для тестування необхідної нейронної мережі.

Продовження таблиці 4.3

Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Технологія створення інтернет-додатків ASP.NET із використанням методів нелінійної оптимізації.	Так.	Дані технології доступні.
	Мова програмування JavaScript із використанням методів нелінійної оптимізації.	Так.	Дані технології доступні, проте написання онлайн-платформи лише на мові JavaScript займе велику кількість часу.
Обрана технологія реалізації ідеї проекту: технологія створення інтернет-додатків ASP.NET із використанням методів нелінійної оптимізації.			

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку. Результати даного аналізу зображено в таблиці 4.4.

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку програмних продуктів для проектування електричних машин	Характеристика
1.	Кількість головних гравців, од	2-3
2.	Загальний обсяг продаж, грн/ум.од	30000
3.	Динаміка ринку (якісна оцінка)	Зростає
4.	Наявність обмежень для входу (вказати характер обмежень)	Програмний продукт із зрозумілим інтерфейсом, здатний проектувати двигуни
5.	Специфічні вимоги до стандартизації та сертифікації	Немає
6.	Середня норма рентабельності в галузі (або по ринку), %	150

Таким чином, за попереднім оцінюванням, ринок є привабливим для входження.

Надалі визначимо потенційні групи клієнтів, їх характеристики, та сформуємо орієнтовний перелік вимог до товару для кожної групи. Ці дані зображено в таблиці 4.5.

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Створення економічно вигідних електричних двигунів.	Компанії з розробки електричних машин та двигунів.	Запропонований програмою двигун має задовольняти певні технічні характеристики, які задані клієнтами.	Програмний продукт має працювати в режимі онлайн, зберігати результати в базі даних компанії.

Продовження таблиці 4.5

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
2.	Створення електричних двигунів потужніших за двигуни внутрішнього згоряння.	Промислові компанії, які використовують автоматизовану зборку продуктів.	Запропонований програмою двигун має задовольняти певні технічні та технологічні характеристики, які задані клієнтами та чинним законодавством.	Програмний продукт має працювати в режимі онлайн, зберігати результати в базі даних компанії та не давати помилки в розрахунках.

Після визначення потенційних груп клієнтів проведемо аналіз ринкового середовища: складемо таблиці факторів, що сприяють ринковому впровадженню проекту (таблиця 4.6), та факторів, що йому перешкоджають (таблиця 4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Недовіра користувача	Недовіра користувача до складних математичних алгоритмів, які використовує програмний продукт, через що він відмовиться купувати дану платформу.	Організація зустрічей із представниками компаній-користувачів та демонстрація ним даних алгоритмів.

Продовження таблиці 4.6

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
2.	Зменшення кількості компаній, які створюють електричні двигуни	Відмова від створення електричних двигунів через їх малу потужність.	Модифікація платформи на створення двигунів інших видів, наприклад, двигунів внутрішнього згоряння.

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Унікальний функціонал програмного продукту	Не існує відомих програмних продуктів, які здатні проектувати електричні двигуни з оптимальними характеристиками.	Нарощувати унікальність програмного продукту за рахунок виходу на ринок інших машин.
2.	Невисока ціна за впровадження та використання програмного забезпечення	За досить невеликі кошти користувач отримує можливість зменшити свої витрати на виробництво майже на 50%.	Збільшувати ціну на програмне забезпечення дуже поступово, щоби було більше користувачів.
3.	Використання ПЗ для електричних двигунів	Можливість використання платформи саме для електричних двигунів, які найбільш екологічними.	Продовжувати підтримувати компанії, які створюють електричні двигуни, для збереження довілля.

Надалі проведемо аналіз пропозиції: визначимо загальні риси конкуренції на ринку. Результати даного аналізу зображені в таблиці 4.8.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції - монополія	Не існує компаній, які за допомогою ПЗ створюють оптимальні двигуни.	Створити працюючий програмний продукт, який не буде давати помилок в розрахунках.
2. За рівнем конкурентної боротьби - національний	Подібні компанії можуть надавати своє ПЗ лише для певного регіону чи країни.	Гарний маркетинг, щоб багато клієнтів по країні дізналися про подібний програмний продукт.
3. За галузевою ознакою - внутрішньогалузевий	Даний продукт призначений лише для виробників електричних двигунів.	Гарний маркетинг, щоб багато клієнтів даної галузі дізналися про подібний програмний продукт.
4. Конкуренція за видами товарів - товарно-видова	Даний продукт працює лише для двигунів, які працюють на електроенергії.	Створити працюючий програмний продукт, який буде використовувати багато технічних характеристик електричних двигунів.
5. За характером конкурентних переваг - нецінова	Головна ідея даного продукту – створення працюючих двигунів, а не отримання коштів з продажу ПЗ.	Створити працюючий програмний продукт, який не буде давати помилок в розрахунках.
6. За інтенсивністю - марочна	Продукт має мати свою марку та товарний знак.	Гарний дизайн товарного знаку, що допоможе в рекламі товару.

Після аналізу конкуренції проведемо більш детальний аналіз умов конкуренції в галузі програмних продуктів для проектування двигунів. Результати даного аналізу зображено в таблиці 4.9.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Динаміка галузі, продуктова лінія, бар'єри проникнення	Наявність товарних знаків, доступ до ресурсів, патенти на продукти	Концентрація постачальників, диференціація витрат	Рівень чутливості до зміни цін, прибутки, контроль якості	Ціна, лояльність споживачів
Висновки:	Конкуренція не є інтенсивною, адже даний ринок ще ніким не зайнятий.	Для входу на ринок необхідно створити товарний знак та написати бета-версію програмного продукту. На даний момент потенційних конкурентів немає.	Постачальники не диктують умови роботи на ринку, бо програмному продукту не потрібно постачання.	Клієнти диктують умови роботи на ринку, бо вони є єдиним джерелом прибутку компанії.	При наявності товарів замінників необхідно буде зменшувати ціну програмного продукту чи створювати ПЗ для інших технічних систем.

Тепер визначимо та обґрунтуємо фактори конкурентоспроможності, які зображені в таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Унікальний функціонал програмного продукту	Не існує програмних продуктів, які використовуються для створення оптимальних двигунів.
2.	Невисока ціна за впровадження та використання ПЗ	Впровадження та використання ПЗ будуть коштувати невисоку ціну, хоча дане ПЗ є монопольним продуктом.
3.	Використання ПЗ для електричних двигунів	В наш час електрична енергія вважається більш екологічною, ніж використання палива. Якщо в майбутньому відбудеться повний перехід на відновлювальні джерела енергії, то даний продукт стане ще більш конкурентоспроможним, адже двигуни будуть ще більш екологічними.

За визначеними факторами конкурентоспроможності проведемо аналіз сильних та слабких сторін стартап-проекту. Результати даного аналізу зображено в таблиці 4.11.

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін онлайн-платформи «EngineOpt»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з «EngineOpt»						
			-3	-2	-1	0	+1	+2	+3
1.	Унікальний функціонал програмного продукту	18		+					
2.	Невисока ціна за використання ПЗ	15			+				
3.	Використання ПЗ для електричних двигунів	16				+			

Тепер проведемо SWOT-аналіз на основі виділених загроз і можливостей, та сильних і слабких сторін проекту. SWOT-матриця зображено в таблиці 4.12.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: унікальність програмного продукту, мала ціна за встановлення та використання ПЗ.	Слабкі сторони: використання не всіх відомих технічних характеристик двигунів, іноді алгоритми математичної оптимізації не дають точний результат.
Можливості: повний перехід людства на відновлювальні джерела електроенергії.	Загрози: відмова від електричних двигунів, бо вони є менш потужними, ніж двигуни внутрішнього згоряння, створення конкурентами схожих програмних продуктів.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтований оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок. Дані альтернативи зображено в таблиці 4.13.

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Створення MVP та впровадження його в невелику кількість компаній машинобудівників	Середня	1 рік
2.	Створення повноцінного продукту, який призначений для будь-яких двигунів	Велика	3 роки
3.	Презентація багатьом компаніям ідеї і повільне створення продукту	Мала	2 роки

Серед даних альтернатив було обрано першу альтернативу, адже строки її реалізації найменші та є ймовірність отримання ресурсів.

4.4 Розроблення ринкової стратегії проекту

Для розроблення ринкової стратегії першим кроком необхідно описати цільові груп потенційних споживачів, які можна побачити в таблиці 4.14.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Підприємства сфери машинобудування	Середня	5-10 підприємств в рік	Середня	Середня
2.	Підприємства сфери легкої промисловості	Не готові	5-10 підприємств в рік	Слабка	Середня
3.	Заклади освіти	Готові	3-5 закладів в рік	Слабка	Проста
Було обрано цільову групу підприємств сфери машинобудування.					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку, якуображено в таблиці 4.15.

Таблиця 4.15 – Визначення базової стратегії розвитку

Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Концентрація на потребах одного цільового сегменту - створенні оптимальних електричних двигунів.	Створений продукт є унікальним і не має аналогів.	Стратегія спеціалізації.

Наступним кроком є вибір стратегії конкурентної поведінки, яку зображено в таблиці 4.16.

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Так.	Компанія буде шукати нових споживачів, але також буде намагатися забирати існуючих у конкурентів.	Компанія не буде копіювати характеристики конкурентів.	Стратегія заняття конкурентної ніші.

Тепер розробимо стратегію позиціонування, що полягає у формуванні ринкової позиції (комплексу асоціацій), за яким споживачі мають ідентифікувати торговельну марку/проект. Її зображено в таблиці 4.17.

Таблиця 4.17 – Визначення стратегії позиціонування

Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
Проектування двигуна має бути швидким та отримані параметри повинні бути в допустимих межах.	Створення оптимальних електричних двигунів.	Товар є «першопрохідцем» на ринку та проект не буде копією інших програм.	Перша в країні програма для оптимального проектування двигунів, одна програма - єдина база спроектованих двигунів, програма працює в режимі онлайн.

4.5 Розроблення маркетингової програми стартап-проекту

Сформуємо маркетингову концепцію товару, який отримає споживач. В таблиці 4.18 зображено результати попереднього аналізу конкурентоспроможності товару.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Створення економічно вигідних електричних двигунів.	Програмний продукт надає змогу користувачеві обрати обмеження на матеріали для створення двигуна.	Можливість вибору ціни майбутнього електричного двигуна.

Продовження таблиці 4.18

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
2.	Створення електричних двигунів потужніших за двигуни внутрішнього згоряння.	Програмний продукт надає змогу користувачеві обрати обмеження на технічні характеристики для створення двигуна.	Можливість вибору значень технічних характеристик майбутнього електричного двигуна.

Надалі розробимо трирівневу маркетингову модель товару: уточнимо ідею продукту, його фізичні складові, особливості процесу його надання. Дана модель зображена в таблиці 4.19.

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові
I. Товар за задумом	Програмний продукт для створення електричних двигунів, який дозволяє користувачу задавати власні обмеження на технічні та технологічні характеристики двигуна та критерій оптимальності, оптимум якого буде розраховано.
II. Товар у реальному виконанні	Властивості / характеристики: 1. Можливість задавати власні технічні характеристики двигуна. 2. Можливість вибору критерію оптимальності серед запропонованих. 3. Зберігання отриманих результатів в базі даних користувача.
	Якість: програмний продукт пройшов всі етапи тестування та готовий до використання.
	Файл з розширенням “.exe” та схема бази даних.
	Марка: назва організації-розробника «iEngine», назва товару «EngineOpt».

Продовження таблиці 4.19

Рівні товару	Сутність та складові
III. Товар із підкріпленням	Спеціаліст із впровадження встановлює ПЗ.
	Відділ розробки підтримує життєдіяльність ПЗ.
Захист програмного продукту буде організовано за допомогою ноу-хау.	

Тепер визначимо цінові межі, якими необхідно керуватись при встановленні ціни на потенційний товар, яке передбачає аналіз ціни на товари-аналоги або товари субститути, а також аналіз рівня доходів цільової групи споживачів. Аналіз проводився експертним методом і його результати зображено в таблиці 4.20.

Таблиця 4.20 – Визначення меж встановлення цін

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
30000-50000 \$/рік	50000-60000 \$/рік	300000-500000 \$/рік	Нижня межа – 40000 \$/рік, верхня межа - 50000 \$/рік

Надалі визначимо оптимальну систему збуту, в межах якого приймається рішення. Дану систему зображено в таблиці 4.21.

Таблиця 4.21 – Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнт виплачує гроші на рік, тоді до нього приходить спеціаліст із впровадження інформаційних систем і встановлює ПЗ на комп'ютер клієнта.	Встановити програмний продукт на комп'ютери клієнтів.	Один посередник – спеціаліст по впровадженню інформаційних систем.	Канал збуту одного рівня.

Тепер розробимо концепцію маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів. Дану концепцію зображено в таблиці 4.22.

Таблиця 4.22 – Концепція маркетингових комунікацій

Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
Клієнт намагається знайти нові методи покращення створюваних ним двигунів.	Е-mail, мережа Інтернет, соціальні мережі.	Унікальність ПЗ, можливість вибору власних критеріїв оптимальності двигуна.	Продемонструвати унікальність даного ПЗ та показати можливу економію щодо створення електричних двигунів при його використанні.	Показати можливість не за велику ціну збільшити свій прибуток за допомогою покращення двигунів.

Висновки до розділу

В даному розділі було повністю виконано перший етап розроблення стартап-проекту, а саме, виконано маркетинговий аналіз стартап-проекту.

За допомогою нього можна сказати, що існує можливість ринкової комерціалізації проекту, адже на ринку програм для підприємств з машинобудування наявний попит на програмні продукти, здатні оптимально проектувати машини, до того ж рентабельність роботи є досить високою.

З огляду на потенційну групу клієнтів, а саме, підприємства сфери машинобудування, та відсутності конкуренції є великі перспективи впровадження даного програмного забезпечення.

Для ринкової реалізації проекту доцільно обрати таку альтернативу впровадження: створення MVP та впровадження його в невелику кількість компаній машинобудівників.

ВИСНОВКИ ПО РОБОТІ І РЕКОМЕНДАЦІЇ ДО ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У даній магістерській дисертації досліджено методи та алгоритми оптимального проектування електричних двигунів загальнопромислового призначення. Було розглянуто основні етапи проектування технічних систем та порівняно апарат оптимізації статичних та динамічних систем.

Нелінійне програмування – один з розділів прикладної математики, який дозволяє створювати статичні системи відповідно до вимог користувача. Тому його було взято за математичну основу даної магістерської дисертації. Було обрано чотири методи нелінійної оптимізації для проектування електричних двигунів та досліджено їх алгоритми для використання на ЕОМ.

Написаний прикладний онлайн-ресурс на мові C#, виконаний за допомогою платформи розробки веб-додатків ASP.NET, дозволяє проводити проектування однофазного асинхронного двигуна із різними технічними та технологічними обмеженнями, а також різними критеріями оптимальності.

Порівняння обраних алгоритмів нелінійної оптимізації показало, що серед методів нелінійної оптимізації найкращим для проектування технічних систем є модифікований метод Лагранжа, адже він дає якісні результати майже з будь-якої точки допустимої області задачі. До того ж, було запропоновано використання в проектуванні двигунів метод комплексів, який належить до методів нульового порядку, який дав також потрібні результати.

Створення нових і модифікація старих методів нелінійної оптимізації відкриває великі можливості в проектуванні статичних систем будь-якої сфери діяльності людини.

ПЕРЕЛІК ПОСИЛАНЬ

1. Хог Э. Прикладное оптимальное проектирование / Э. Хог, Я. Арора ; пер. с англ. – М. : Мир, 1983. – 478 с.
2. Антонов М. В. Технология производства электрических машин / М. В. Антонов, Л. С. Герасимов. – М. : Энергоизда, 1982. – 512 с.
3. Optimum Design of I. C. Engine Pistons [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: <http://mechanicaldesign.asmedigitalcollection.asme.org/article.aspx?articleid=1452325>.
4. Optimal design of thermoacoustic engines driven by LNG using particle swarm optimization [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: https://www.jstage.jst.go.jp/article/transjsme/advpub/0/advpub_16-00248/_article.
5. Акулич И. Л. Математическое программирование в примерах и задачах: Учеб. пособие для студентов эконом. спец. вузов / Иван Людвигович Акулич. – М. : Высшая школа, 1986. – 319 с.
6. Young K. A. Optimal design of engine mount using an artificial life algorithm / Kong Ahn Young, Jin Dae Song, Bo-SukYang // Journal of Sound and Vibration. – 2003. – No. 261. – pp. 309-328.
7. Fellini R. Quantitative platform selection in optimal design of product families, with application to automotive engine design / Ryan Fellini, Michael Kokkolaras, Panos Y. Papalambros // Journal of Engineering Design. – 2006. – No. 17. – pp. 429-446.
8. Optimal design of Stirling heat engine using an advanced optimization algorithm [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <https://www.ias.ac.in/article/fulltext/sadh/041/11/1321-1331>.
9. Kobayashi H. Optimal Design of Hypersonic Turbojet Engines for Two-Stage-to-Orbit Spaceplane / Hiroaki Kobayashi, Hirokazu Suzuki // Journal of Spacecraft and Rockets. – 2008. – No. 45. – pp. 741-749.

10. Однофазный асинхронный двигатель: принцип работы [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://dovidkam.com/remont/elektrika/odnofaznij-asinxronnij-dvigun-princip-roboti.html>.
11. Лопухина Е. М. Проектирование асинхронных микродвигателей с применением ЭВМ / Е. М. Лопухина, Г. А. Семенчуков. – М. : Высшая школа, 1980. – 512 с.
12. Адаменко А. И., Преобразование однофазного тока в многофазный / А. И. Адаменко, В. И. Кисленко. – К. : Техніка, 1971. – 271 с.
13. Optimal Design of a Solar-Driven Heat Engine Based on Thermal and Ecological Criteria [Электронный ресурс]. – 2015. – Режим доступа до ресурсу: <http://ascelibrary.org/doi/abs/10.1061/%28ASCE%29EY.1943-7897.0000191>.
14. Egorov I. N. Optimal design and control of gas-turbine engine components: a multicriteria approach / I.N. Egorov, G.V. Kretinin, I.A. Leshchenko // MCB UP Ltd. – 1997. – No. 69. – pp. 518-526.
15. Optimal Design for a Highly Downsized Gasoline Engine [Электронный ресурс]. – 2009. – Режим доступа до ресурсу: <https://www.sae.org/publications/technical-papers/content/2009-01-1794>.
16. Реклейтис Г. Оптимизация в технике : в 2 кн. / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел; пер. с англ. – М. : Мир. – Кн. 1. – 1986. – 349 с.
17. Зангвилл У. Нелинейное программирование. Единый подход. / У. Зангвилл; пер. с англ. Е. Г. Гольштейна. – М. : Советское радио, 1973. – 312 с.
18. Карманов В. Г. Математическое программирование / Владимир Георгиевич Карманов. – М. : Наука. Главная редакция физико-математической литературы, 1980. – 256 с.
19. Зайченко Ю. П. Исследование операций / Юрий Петрович Зайченко. – К. : Вища школа, 1975. – 816 с.
20. Bazaraa M. S. Nonlinear Programming: Theory and Algorithms / M. S. Bazaraa, C. M. Shetty. – N.Y. : Wiley, 1979. – 862 p.

21. Han S. P. Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems / S. P. Han // Mathematical Programming. – 1976. – No. 11. – pp. 263-282.
22. Реклейтис Г. Оптимизация в технике : в 2 кн. / Г. Реклейтис, А. Рейвиндран, К. Рэгсдел; пер. с англ. – М. : Мир. – Кн. 2. – 1986. – 320 с.
23. Siddal J. N. Analytical Decision-Making in Engineering Design / J. N. Siddal. – Englewood Cliffs : Prentice-Hall, 1972. – 431 p.
24. Box M. J. Nonlinear Optimization Techniques / M. J. Box, D. Davies, W. H. Swann. – London : Oliver & Boyd, 1969. – 60 p.
25. Руководство по разработке для .NET Framework [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: [https://msdn.microsoft.com/ru-ru/library/hh156542\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/hh156542(v=vs.110).aspx).
26. Полное руководство по языку программирования C# 7.0 и платформе .NET 4.7 [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://metanit.com/sharp/tutorial>.

ДОДАТОК А ЛІСТИНГ ПРОГРАМИ

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Windows;
using System.ComponentModel.DataAnnotations;
using System.Globalization;
using System.Threading;
using System.Configuration;

namespace Диссертация.Annotations
{
    public class CorrectDateAttribute :
        ValidationAttribute
    {
        private static DateTime db;

        public override bool IsValid(object value)
        {
            if (value != null)
            {
                string strval = value.ToString();
                DateTime.TryParse(strval, out db);
                if (db >= new DateTime(1900, 1, 1))
                    return true;
                else
                    return false;
            }
            return true;
        }
        public override string
        FormatErrorMessage(string name)
        {
            var culture =
                ConfigurationManager.AppSettings.Get("Culture");
            return
                String.Format(Thread.CurrentThread.CurrentCulture,
                    ErrorMessageString, name);
        }
    }
}

-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Text.RegularExpressions;

namespace Оптимальное_проектирование_двигателей
{
    public class Complex
    {
        double re;
        double im;

        public double Re
        {
            get { return re; }
            set { re = value; }
        }

        public double Im
        {
            get { return im; }
            set { im = value; }
        }

        public Complex(double real_part, double
        imaginary_part)
        {
            Re = real_part;
            Im = imaginary_part;
        }

        public static Complex operator +(Complex a,
        Complex b)
        {
            return new Complex(a.Re + b.Re, a.Im +
        b.Im);
        }

        public static Complex operator +(Complex a,
        double b)
        {
            return new Complex(a.Re + b, a.Im);
        }

        public static Complex operator -(Complex a,
        Complex b)
        {
            return new Complex(a.Re - b.Re, a.Im -
        b.Im);
        }

        public static Complex operator -(Complex a,
        double b)
        {
            return new Complex(a.Re - b, a.Im);
        }

        public static Complex operator -(double a,
        Complex b)
        {
            return new Complex(a - b.Re, -b.Im);
        }

        public static Complex operator -(Complex a)
        {
            return new Complex(-a.Re, -a.Im);
        }

        public static Complex operator *(Complex a,
        Complex b)
        {
            return new Complex(a.Re * b.Re - a.Im *
        b.Im,
                a.Im * b.Re + a.Re * b.Im);
        }

        public static Complex operator *(Complex a,
        double d)
        {
            return new Complex(d * a.Re, d * a.Im);
        }
    }
}

```

```

    }

    public static Complex operator *(double d,
Complex a)
    {
        return new Complex(d * a.Re, d * a.Im);
    }

    public static Complex operator /(Complex a,
Complex b)
    {
        return a * Conj(b) * (1 / (Abs(b) *
Abs(b)));
    }

    public static Complex operator /(Complex a,
double b)
    {
        return a * (1 / b);
    }

    public static Complex operator /(double a,
Complex b)
    {

```

```

        return a * Conj(b) * (1 / (Abs(b) *
Abs(b)));
    }

    public static double Abs(Complex a)
    {
        return Math.Sqrt(a.Im * a.Im + a.Re *
a.Re);
    }

    public static Complex Inv(Complex a)
    {
        return new Complex(a.Re / (a.Re * a.Re +
a.Im * a.Im),
        -a.Im / (a.Re * a.Re + a.Im * a.Im));
    }

    public static Complex Conj(Complex a)
    {
        return new Complex(a.Re, -a.Im);
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using MathNet.Numerics;
using MathNet.Numerics.LinearAlgebra;

namespace Диссертация.Calculation
{
    public class Functions
    {
        public static Func<Vector<double>, double>
FindFunction(int n)
        {
            switch (n)
            {
                case 1:
                    return Functions.Mmax;
                case 2:
                    return Functions.Mt;
                case 3:
                    return Functions.Mc;
                case 4:
                    return Functions.Gff;
                case 5:
                    return Functions.Cosfi;
                case 6:
                    return Functions.Eff;
                case 7:
                    return Functions.M;
                case 8:
                    return Functions.Tw;
                default:
                    return Functions.Mmax;
            }
        }

        public static double Mmax(Vector<double> x)
// Кратность максимального момента
        {
            const double pi = Math.PI;

            double Di = x[0]; //диаметр расточки
статора
            double li = x[1]; //длина пакета статора
статора
            double bzs = x[2]; // ширина зубца

```

```

            double bzs = x[2]; // ширина зубца ротора
            double has = x[4]; //высота стенки
статора
            double har = x[5]; //высота стенки ротора
            double Da = x[6]; // наружный диаметр
листа статора

            double Pn = 250, p1 = 2, f1 = 50, m1 = 2,
Cp = 10, Zs = 24, Zr = 17, dv = 0.02, w2k2 = 0.5, dg1
= 0.00063, dg2 = 0.0005;
            double ro75 = 0.000001 / 46.0, del =
0.00025, deliz = 0.00019, kpd1 = 0.6, roh = 0.000001
/ 22.5, Up1 = 112, Up2 = 162;
            double mu0 = 4.0 * pi * 0.0000001, kmu1 =
1.48, kc = 0.95, Kts = 0.9, Ktr = 1.3, lampr = 1.39,
lamlr = 0.2;

            Complex X = new Complex();
            Complex Y = new Complex();
            Complex Zpr = new Complex();
            Complex V = new Complex();
            Complex T = new Complex();
            Complex Zdd = new Complex();
            Complex Ic = new Complex();
            Complex T5 = new Complex();
            Complex Zabr = new Complex();
            Complex W = new Complex();
            Complex Y1 = new Complex();
            Complex T1 = new Complex();
            Complex Zqq = new Complex();
            Complex Zqd = new Complex();
            Complex Zdq = new Complex();
            Complex Y2 = new Complex();
            Complex T2 = new Complex();
            Complex U1 = new Complex();
            Complex T3 = new Complex();
            Complex T4 = new Complex();
            Complex T6 = new Complex();
            Complex T7 = new Complex();
            Complex T8 = new Complex();
            Complex T9 = new Complex();
            Complex Iq = new Complex();
            Complex Id = new Complex();
            Complex T14 = new Complex();
            Complex k = new Complex();
            Complex T10 = new Complex();
            Complex T11 = new Complex();

```

```

Complex T12 = new Complex();
Complex T13 = new Complex();
Complex Apr1 = new Complex();
Complex wkdl = new Complex();
Complex wkql = new Complex();
Complex K1 = new Complex();
Complex k2 = new Complex();
Complex Aabr1 = new Complex();
Complex T15 = new Complex();
Complex T16 = new Complex();
Complex T17 = new Complex();
Complex T18 = new Complex();
Complex T19 = new Complex();
Complex T20 = new Complex();
Complex K3 = new Complex();
double alfs1 = 2 * pi * p1 / Zs;
double alfr1 = 2 * pi * p1 / Zr;
double qd = Zs / (2 * p1 * m1);
Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
double Kq = Kd;
double Tetd1 = -2.618, Tetq1 = -1.047;
double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
* Math.Sin(alfr1 / 2) / alfr1;
double kt = 6;
double n1 = 60 * f1 / p1;
double Pmex = kt * Math.Pow(n1 / 1000, 2)
* Math.Pow(Da * 10, 3);
double Pdab = 0.005 * Pn / kpd1, Prach =
Pn + Pmex + Pdab, Mt = 9.55 * (Pmex + Pdab) / n1;
double Tay = (pi * Di) / (2 * p1), ts =
pi * Di / Zs;
double Dr = Di - 2 * del, tr = pi * Dr /
Zr, lr = li;
double bhs = 0.002, hhs = 0.0005;
double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
double hps = (Da - Di - 2 * has) / 2;
double h12s = hps - hhs - d1s / 2;
double bhr = 0.001, hhr = 0.0007;
double d1r = (pi * (Dr - 2 * hhr) - bzs *
Zr) / (Zr + pi);
double hpr = d1r + hhr, Spr = pi *
Math.Pow(d1r, 2) / 4;
double Skl = Spr * 0.625 / Math.Sin(alfr1
/ 2), bkl = 0.009;
double akl = Skl / bkl, Dkl = Dr - bkl;
double bck = 1, Gamck = 2 * bck * ts /
Dr;
double kck = 2 * Math.Sin(p1 * Gamck / 2)
/ (p1 * Gamck);
double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);
double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;
double K0 = 1.55, B0 = 1, Ysr = 11.741 /
3.0;
double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = K0 * TayY + 0.02 * B0;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hkl = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hkl - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;

double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmu1), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dkl / (Zr * akl * bkl);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +
x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);

double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
Spr = 0.0885;
double Sabr = 2 - 0.0885;
X.Re = r11;
X.Im = x11;
Y.Re = r21h;
Y.Im = Spr * x21h;
V.Re = Spr * r11 + r21h;
V.Im = Spr * (x11 + x21h);
Y1.Re = r21h;
Y1.Im = Sabr * x21h;
W.Re = Sabr * r11 + r21h;
W.Im = Sabr * (x11 + x21h);
T = X * Y;
Zpr = T / V;
T1 = X * Y1;
Zabr = T1 / W;
Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);
Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);
Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
Zabr.Im);
Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
Zabr.Re);

Y2.Re = gc;
Y2.Im = bc;
U1.Re = 220;
U1.Im = 0;
k.Re = 1;
k.Im = 0;
T2 = Zdd - Zqd;
T3 = T2 * Y2;
T4 = T3 * U1;
T5 = Zdd * Zqq;
T6 = Zdq * Zqd;
T7 = T5 - T6;
T8 = T7 * Y2;
T9 = Zdd + T8;
Iq = T4 / T9;
T10 = Zqq - Zdq;
T11 = T10 * Y2;
T12 = T11 + k;

```

```

T13 = T12 * U1;
T14 = Zdd + T8;
Id = T13 / T14;
wkd1.Re = 672 * 0.9106;
wkd1.Im = 0;
wkq1.Re = 972 * 0.9106;
wkq1.Im = 0;
K1.Re = 0;
K1.Im = -1;
k2.Re = 0;
k2.Im = 1;
T15 = Id * wkd1;
T16 = Iq * wkq1;
T17 = T16 * K1;
Apr1 = T15 + T17;
double Apr11Apr11 =
Math.Pow(Complex.Abs(Apr1), 2);
T18 = T16 * k2;
Aabr1 = T15 + T18;
double Aabr11Aabr11 =
Math.Pow(Complex.Abs(Aabr1), 2);
double Alfd = 0.6782, kb = 1.117;
T19 = Zpr * Apr1;
T20 = Zabr * Aabr1;
K3 = T19 + T20;
double K33 = Complex.Abs(K3);
double Fr = K33 / (4 * kb * f1);
double K44 = Complex.Abs(T19);
double Fpr = K44 / (4 * kb * f1);
double K55 = Complex.Abs(T20);
double Fabr = K55 / (4 * kb * f1);
double Bdel = Fr / (Alfd * Tay * li),
Bdelpr = Fpr / (Alfd * Tay * li), Bdelabr = Fabr /
(Alfd * Tay * li);
double Bzsh = Bdel * (ts / (bzs * kc)),
Bzshpr = Bdelpr * (ts / (bzs * kc)), Bzshabr =
Bdelabr * (ts / (bzs * kc));
double Bzrh = Bdel * (tr / (bzh * kc)),
Bzrhpr = Bdelpr * (tr / (bzh * kc)), Bzrhabr =
Bdelabr * (tr / (bzh * kc));
double Bas = Fr / (2 * li * has * kc),
Baspr = Fpr / (2 * li * has * kc), Basabr = Fabr / (2
* li * has * kc);
double Bar = Fr / (2 * lr * har * kc),
Barpr = Fpr / (2 * lr * har * kc), Barabr = Fabr / (2
* lr * har * kc);
double p_10_50 = 3.3, Gas = 7800.0 * pi *
(Da - has) * has * li * kc;
double Paspr = 1.6 * p_10_50 *
Math.Pow(Baspr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
double Pasabr = 1.6 * p_10_50 *
Math.Pow(Basabr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
double Gzs = 7800.0 * Zs * bzs * hps * li
* kc;
double Pzspr = 1.8 * p_10_50 *
Math.Pow(Bzshpr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
double Pzsabr = 1.8 * p_10_50 *
Math.Pow(Bzshabr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
double Gar = 7800.0 * pi * (dv + har) *
har * li * kc;
double Parpr = 1.6 * p_10_50 *
Math.Pow(Barpr, 2) * Gar * Math.Pow(f1 * Spr / 50,
1.3);
double Parabr = 1.6 * p_10_50 *
Math.Pow(Barabr, 2) * Gar * Math.Pow(f1 * Sabr / 50,
1.3);
double Gzr = 7800.0 * Zr * bzh * hpr * li
* kc;
double Pzrpr = 1.8 * p_10_50 *
Math.Pow(Bzrhpr, 2) * Gzr * Math.Pow(f1 * Spr / 50,
1.3) * Ktr;

```

```

double Pzrabr = 1.8 * p_10_50 *
Math.Pow(Bzrhabr, 2) * Gzr * Math.Pow(f1 * Sabr / 50,
1.3) * Ktr;
double Pctabr = Pasabr + Pzsabr + Parabr
+ Pzrabr, Pctpr = Paspr + Pzspr + Parpr + Pzrpr;
double r11pr1 = 0.1820 * 0.0001, r11abr1
= 0.334 * 0.0001;
double Zmupr = Math.Sqrt(Math.Pow(r11pr1,
2) + Math.Pow(x11, 2));
double r11pr = Pctpr * Math.Pow(Zmupr, 2)
/ Math.Pow(K44, 2);
double Zmuabr =
Math.Sqrt(Math.Pow(r11abr1, 2) + Math.Pow(x11, 2));
double r11abr = Pctabr * Math.Pow(Zmuabr,
2) / Math.Pow(K55, 2);
double Apr11 = Math.Sqrt(Apr11Apr11),
Aabr11 = Math.Sqrt(Aabr11Aabr11);
double Ppsipr = Math.Pow(Apr11, 2) *
((Math.Pow(r11pr, 2) + Math.Pow(x11, 2)) * r21h *
Spr) / (Math.Pow(r11pr * Spr + r21h, 2) +
Math.Pow(Spr, 2) * Math.Pow(x11 + x21h, 2));
double Ppsiabr = Math.Pow(Aabr11, 2) *
((Math.Pow(r11abr, 2) + Math.Pow(x11, 2)) * r21h *
Sabr) / (Math.Pow(r11abr * Sabr + r21h, 2) +
Math.Pow(Sabr, 2) * Math.Pow(x11 + x21h, 2));
double Mpr = (p1 * Ppsipr) / (2 * pi *
f1), Mabr = (p1 * Ppsiabr) / (2 * pi * f1);
double Mres = Mpr - Mabr;
double Pemmax = 475.674;
double Mmax = p1 * Pemmax / (2 * pi *
f1);
double M2 = Mres - Mt;
return Mmax / M2;
}

```

```

public static double Mt(Vector<double> x) //
Кратность начального пускового момента
{
    const double pi = Math.PI;
    double Di = x[0]; //диаметр расточки
    double li = x[1]; //длина пакета статора
    double bzs = x[2]; // ширина зубца
    double bzh = x[3]; // ширина зубца ротора
    double has = x[4]; //высота стенки
    double har = x[5]; //высота стенки ротора
    double Da = x[6]; // наружный диаметр
    листа статора
    double Pn = 250, U = 220, p1 = 2, f1 =
    50, m1 = 2, Cp = 10, Zs = 24, Zr = 17, dv = 0.02,
    w2k2 = 0.5, dg1 = 0.00063, dg2 = 0.0005;
    double ro75 = 0.000001 / 46.0, del =
    0.00025, deliz = 0.00019, kpd1 = 0.6, roh = 0.000001
    / 22.5, Up1 = 112, Up2 = 162;
    double mu0 = 4.0 * pi * 0.0000001, kmu1 =
    1.48, kc = 0.95, Kts = 0.9, Ktr = 1.3, lampr = 1.39,
    lamlr = 0.2;
    Complex X = new Complex();
    Complex Y = new Complex();
    Complex Zpr = new Complex();
    Complex V = new Complex();
    Complex T = new Complex();
    Complex Zdd = new Complex();
    Complex Ic = new Complex();
    Complex T5 = new Complex();
    Complex Zabr = new Complex();
    Complex W = new Complex();
    Complex Y1 = new Complex();
    Complex T1 = new Complex();
}

```

```

Complex Zqq = new Complex();
Complex Zqd = new Complex();
Complex Zdq = new Complex();
Complex Y2 = new Complex();
Complex T2 = new Complex();
Complex U1 = new Complex();
Complex T3 = new Complex();
Complex T4 = new Complex();
Complex T6 = new Complex();
Complex T7 = new Complex();
Complex T8 = new Complex();
Complex T9 = new Complex();
Complex Iq = new Complex();
Complex Id = new Complex();
Complex T14 = new Complex();
Complex k = new Complex();
Complex T10 = new Complex();
Complex T11 = new Complex();
Complex T12 = new Complex();
Complex T13 = new Complex();
Complex Apr1 = new Complex();
Complex wkdl = new Complex();
Complex wkq1 = new Complex();
Complex K1 = new Complex();
Complex k2 = new Complex();
Complex Aabr1 = new Complex();
Complex T15 = new Complex();
Complex T16 = new Complex();
Complex T17 = new Complex();
Complex T18 = new Complex();
Complex T19 = new Complex();
Complex T20 = new Complex();
Complex K3 = new Complex();
double alfs1 = 2 * pi * p1 / Zs;
double alfr1 = 2 * pi * p1 / Zr;
double qd = Zs / (2 * p1 * m1);
double Kd = Math.Sin(qd * alfs1 / 2) *
Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
double Kq = Kd;
double Tetd1 = -2.618, Tetq1 = -1.047;
double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
* Math.Sin(alfr1 / 2) / alfr1;
double kt = 6;
double n1 = 60 * f1 / p1;
double Pmex = kt * Math.Pow(n1 / 1000, 2)
* Math.Pow(Da * 10, 3);
double Pdad = 0.005 * Pn / kpd1, Prach =
Pn + Pmex + Pdad, Mt = 9.55 * (Pmex + Pdad) / n1;
double Tay = (pi * Di) / (2 * p1), ts =
pi * Di / Zs;
double Dr = Di - 2 * del, tr = pi * Dr /
Zr, lr = li;
double bhs = 0.002, hhs = 0.0005;
double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
double hps = (Da - Di - 2 * has) / 2;
double h12s = hps - hhs - d1s / 2;
double bhr = 0.001, hhr = 0.0007;
double d1r = (pi * (Dr - 2 * hhr) - bzs *
Zr) / (Zr + pi);
double hpr = d1r + hhr, Spr = pi *
Math.Pow(d1r, 2) / 4;
double Skl = Spr * 0.625 / Math.Sin(alfr1
/ 2), bkl = 0.009;
double akl = Skl / bkl, Dkl = Dr - bkl;
double bck = 1, Gamck = 2 * bck * ts /
Dr;
double kck = 2 * Math.Sin(p1 * Gamck / 2)
/ (p1 * Gamck);
double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);
double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;

double KO = 1.55, B0 = 1, Ysr = 11.741 /
3.0;
double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = KO * TayY + 0.02 * B0;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hkl = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hkl - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;
double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmul), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dkl / (Zr * akl * bkl);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +
x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);
double x11o = x11 / (x11 + x21h), x21ho =
x21h / (x11 + x21h);
double r11o = r11 / (x11 + x21h), r21ho =
r21h / (x11 + x21h);
double a = 2 * r21h * (r11o * (r11o +
r21ho) - x11o * (1 + x21ho));
double b = 2 * r11o * (1 + x11o) * x21h;
double c = 2 * r21h * (r11o * (1 + x21ho)
+ x11o * (r11o + r21ho));
double d = 2 * x21h * (x11o -
Math.Pow(r11o, 2));
double a1 = r21ho * (r21ho + 2 * r11o),
b1 = Math.Pow(r11o, 2) - 1;
double c1 = 2 * r21ho, d1 = 2 * r11o;
double m = r21h * (r21h * (Math.Pow(r11o,
2) - Math.Pow(x11o, 2)) - 4 * r11o * x11o * x21h);
double n = Math.Pow(x21h, 2) *
(Math.Pow(r11o, 2) - Math.Pow(x11o, 2));
double p = 2 * ((Math.Pow(r11o, 2) -
Math.Pow(x11o, 2)) * x21h + r11o * x11o * r21h) *
r21h;
double Q = 2 * r11o * x11o *
Math.Pow(x21h, 2);
double Ah = (rd * rq - xd * xq) * a1 -
(rd * xq + rq * xd) * c1 + Math.Pow(wd * Kd, 2) * (rq
* a - xq * c) +
Math.Pow(wq * Kq, 2) * (rd * a - xd *
c) + 4 * Math.Pow(wd * Kd * wq * Kq, 2) * m;
double Bh = (rd * rq - xd * xq) * b1 -
(rd * xq + rq * xd) * d1 + Math.Pow(wd * Kd, 2) * (xq
* d - rq * b) +
Math.Pow(wq * Kq, 2) * (xd * d - rd *
b) - 4 * Math.Pow(wd * Kd * wq * Kq, 2) * n;

```



```

double Ch = (rd * rq - xd * xq) * c1 +
(rd * xq + rq * xd) * a1 + Math.Pow(wd * Kd, 2) * (rq
* c + xq * a) +
Math.Pow(wq * Kq, 2) * (rd * c + xd *
a) + 4 * Math.Pow(wd * Kd * wq * Kq, 2) * p;
double Dh = (rd * rq - xd * xq) * d1 +
(rd * xq + rq * xd) * b1 - Math.Pow(wd * Kd, 2) * (rq
* d + xq * b) -
Math.Pow(wq * Kq, 2) * (rd * d + xd *
b) - 4 * Math.Pow(wd * Kd * wq * Kq, 2) * Q;
double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
double AA = rd * a1 - xd * c1 +
Math.Pow(wd * Kd, 2) * a + Ah * gc - Ch * bc;
double BB = rd * b1 - xd * d1 -
Math.Pow(wd * Kd, 2) * b + Bh * gc - Dh * bc;
double CC = rd * c1 + xd * a1 +
Math.Pow(wd * Kd, 2) * c + Ch * gc + Ah * bc;
double DD = rd * d1 + xd * b1 -
Math.Pow(wd * Kd, 2) * d + Dh * gc + Bh * bc;
double Aq = rq * a1 - xq * c1 +
Math.Pow(wq * Kq, 2) * a;
double Bq = rq * b1 - xq * d1 -
Math.Pow(wq * Kq, 2) * b;
double Cq = rq * c1 + xq * a1 +
Math.Pow(wd * Kd, 2) * c; //ошибка в дипломе
(наверное, должно быть wq и Kq)
double Dq = rq * d1 + xq * b1 -
Math.Pow(wq * Kq, 2) * d;
double Ad = rd * a1 - xd * c1 +
Math.Pow(wd * Kd, 2) * a;
double Bd = rd * b1 - xd * d1 -
Math.Pow(wd * Kd, 2) * b;
double Cd = rd * c1 + xd * a1 +
Math.Pow(wd * Kd, 2) * c;
double Ddd = rd * d1 + xd * b1 -
Math.Pow(wd * Kd, 2) * d;
double A11 = wd * Kd * (a1 + Aq * gc - Cq
* bc) + wq * Kq * (Cd * gc + Ad * bc);
double B11 = wd * Kd * (b1 + Bq * gc - Dq
* bc) + wq * Kq * (Ddd * gc + Bd * bc);
double C11 = wd * Kd * (c1 + Cq * gc + Aq
* bc) - wq * Kq * (Ad * gc - Cd * bc);
double D11 = wd * Kd * (d1 + Dq * gc + Bq
* bc) - wq * Kq * (Bd * gc - Ddd * bc);
double A22 = wd * Kd * (a1 + Aq * gc - Cq
* bc) - wq * Kq * (Cd * gc + Ad * bc);
double B22 = wd * Kd * (b1 + Bq * gc - Dq
* bc) - wq * Kq * (Ddd * gc + Bd * bc);
double C22 = wd * Kd * (c1 + Cq * gc + Aq
* bc) + wq * Kq * (Ad * gc - Cd * bc);
double D22 = wd * Kd * (d1 + Dq * gc + Bq
* bc) + wq * Kq * (Bd * gc - Ddd * bc);
double Mp = ((pi * Math.Pow(U, 2)) / (2 *
pi * f1)) * ((Math.Pow(A11 + B11, 2) + Math.Pow(C11 +
D11, 2) - Math.Pow(A22 + B22, 2) - Math.Pow(C22 +
D22, 2)) /
(Math.Pow(AA + BB, 2) + Math.Pow(CC +
DD, 2)) * (((Math.Pow(r11o, 2) + Math.Pow(x11o, 2)) *
r21h) / (Math.Pow(r11o + r21ho, 2) + 1)));
Spr = 0.0885;
double Sabr = 2 - 0.0885;
X.Re = r11;
X.Im = x11;
Y.Re = r21h;
Y.Im = Spr * x21h;
V.Re = Spr * r11 + r21h;
V.Im = Spr * (x11 + x21h);
Y1.Re = r21h;
Y1.Im = Sabr * x21h;
W.Re = Sabr * r11 + r21h;
W.Im = Sabr * (x11 + x21h);
T = X * Y;
Zpr = T / V;
T1 = X * Y1;

Zabr = T1 / W;
Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);
Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);
Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
Zabr.Im);
Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
Zabr.Re);
Y2.Re = gc;
Y2.Im = bc;
U1.Re = 220;
U1.Im = 0;
k.Re = 1;
k.Im = 0;
T2 = Zdd - Zqd;
T3 = T2 * Y2;
T4 = T3 * U1;
T5 = Zdd * Zqq;
T6 = Zdq * Zqd;
T7 = T5 - T6;
T8 = T7 * Y2;
T9 = Zdd + T8;
Iq = T4 / T9;
T10 = Zqq - Zdq;
T11 = T10 * Y2;
T12 = T11 + k;
T13 = T12 * U1;
T14 = Zdd + T8;
Id = T13 / T14;
wkd1.Re = 672 * 0.9106;
wkd1.Im = 0;
wkq1.Re = 972 * 0.9106;
wkq1.Im = 0;
K1.Re = 0;
K1.Im = -1;
k2.Re = 0;
k2.Im = 1;
T15 = Id * wkd1;
T16 = Iq * wkq1;
T17 = T16 * K1;
Apr1 = T15 + T17;
double Apr11Apr11 =
Math.Pow(Complex.Abs(Apr1), 2);
T18 = T16 * k2;
Aabr1 = T15 + T18;
double Aabr11Aabr11 =
Math.Pow(Complex.Abs(Aabr1), 2);
double Alfd = 0.6782, kb = 1.117;
T19 = Zpr * Apr1;
T20 = Zabr * Aabr1;
K3 = T19 + T20;
double K33 = Complex.Abs(K3);
double Fr = K33 / (4 * kb * f1);
double K44 = Complex.Abs(T19);
double Fpr = K44 / (4 * kb * f1);
double K55 = Complex.Abs(T20);
double Fabr = K55 / (4 * kb * f1);
double Bdel = Fr / (Alfd * Tay * li),
Bdelpr = Fpr / (Alfd * Tay * li), Bdelabr = Fabr /
(Alfd * Tay * li);
double Bzsh = Bdel * (ts / (bzs * kc)),
Bzshpr = Bdelpr * (ts / (bzs * kc)), Bzshabr =
Bdelabr * (ts / (bzs * kc));
double Bzrh = Bdel * (tr / (bzh * kc)),
Bzrhpr = Bdelpr * (tr / (bzh * kc)), Bzrhabr =
Bdelabr * (tr / (bzh * kc));

```

```

        double Bas = Fr / (2 * li * has * kc),
Baspr = Fpr / (2 * li * has * kc), Basabr = Fabr / (2
* li * has * kc);
        double Bar = Fr / (2 * lr * har * kc),
Barpr = Fpr / (2 * lr * har * kc), Barabr = Fabr / (2
* lr * har * kc);
        double p_10_50 = 3.3, Gas = 7800.0 * pi *
(Da - has) * has * li * kc;
        double Paspr = 1.6 * p_10_50 *
Math.Pow(Baspr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
        double Pasabr = 1.6 * p_10_50 *
Math.Pow(Basabr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
        double Gzs = 7800.0 * Zs * bzs * hps * li
* kc;
        double Pzspr = 1.8 * p_10_50 *
Math.Pow(Bzshpr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
        double Pzsabr = 1.8 * p_10_50 *
Math.Pow(Bzshabr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
        double Gar = 7800.0 * pi * (dv + har) *
har * li * kc;
        double Parpr = 1.6 * p_10_50 *
Math.Pow(Barpr, 2) * Gar * Math.Pow(f1 * Spr / 50,
1.3);
        double Parabr = 1.6 * p_10_50 *
Math.Pow(Barabr, 2) * Gar * Math.Pow(f1 * Sabr / 50,
1.3);
        double Gzr = 7800.0 * Zr * bzs * hpr * li
* kc;
        double Pzrpr = 1.8 * p_10_50 *
Math.Pow(Bzrhpr, 2) * Gzr * Math.Pow(f1 * Spr / 50,
1.3) * Ktr;
        double Pzrabr = 1.8 * p_10_50 *
Math.Pow(Bzrhabr, 2) * Gzr * Math.Pow(f1 * Sabr / 50,
1.3) * Ktr;
        double Pctabr = Pasabr + Pzsabr + Parabr
+ Pzrabr, Pctpr = Paspr + Pzspr + Parpr + Pzrpr;
        double r11pr1 = 0.1820 * 0.0001, r11abr1
= 0.334 * 0.0001;
        double Zmupr = Math.Sqrt(Math.Pow(r11pr1,
2) + Math.Pow(x11, 2));
        double r11pr = Pctpr * Math.Pow(Zmupr, 2)
/ Math.Pow(K44, 2);
        double Zmuabr =
Math.Sqrt(Math.Pow(r11abr1, 2) + Math.Pow(x11, 2));
        double r11abr = Pctabr * Math.Pow(Zmuabr,
2) / Math.Pow(K55, 2);
        double Apr11 = Math.Sqrt(Apr11Apr11),
Aabr11 = Math.Sqrt(Aabr11Aabr11);
        double Ppsipr = Math.Pow(Apr11, 2) *
((Math.Pow(r11pr, 2) + Math.Pow(x11, 2)) * r21h *
Spr) / (Math.Pow(r11pr * Spr + r21h, 2) +
Math.Pow(Spr, 2) * Math.Pow(x11 + x21h, 2));
        double Ppsiabr = Math.Pow(Aabr11, 2) *
((Math.Pow(r11abr, 2) + Math.Pow(x11, 2)) * r21h *
Sabr) / (Math.Pow(r11abr * Sabr + r21h, 2) +
Math.Pow(Sabr, 2) * Math.Pow(x11 + x21h, 2));
        double Mpr = (p1 * Ppsipr) / (2 * pi *
f1), Mabr = (p1 * Ppsiabr) / (2 * pi * f1);
        double Mres = Mpr - Mabr;
        double M2 = Mres - Mt;

        return Mp / M2;
    }

    public static double Mc(Vector<double> x) //
Кратность начального пускового тока
    {
        const double pi = Math.PI;

        double Di = x[0]; //диаметр расточки
статора
        double li = x[1]; //длина пакета статора

```

```

        double bzs = x[2]; // ширина зубца
статора
        double bzs = x[3]; // ширина зубца ротора
        double has = x[4]; //высота стенки
статора
        double Da = x[6]; // наружный диаметр
листа статора
        /*double Di = 0.072; //диаметр расточки
статора
        double li = 0.048; //длина пакета статора
        double bzs = 0.00464; // ширина зубца
статора
        double bzs = 0.00632; // ширина зубца
ротора
        double has = 0.01; //высота стенки
статора
        double Da = 0.12; // наружный диаметр
листа статора*/

        double U = 220, p1 = 2, f1 = 50, m1 = 2,
Cp = 10, Zs = 24, Zr = 17, w2k2 = 0.5, dg1 = 0.00063,
dg2 = 0.0005, ro75 = 0.000001 / 46.0;
        double del = 0.00025, deliz = 0.00019,
roh = 0.000001 / 22.5, mu0 = 4 * pi * 0.000001, kmu1
= 1.48, lampr = 1.39, lamlr = 0.2;
        double Up1 = 112, Up2 = 162;

        Complex X = new Complex();
        Complex Y = new Complex();
        Complex Zpr = new Complex();
        Complex V = new Complex();
        Complex T = new Complex();
        Complex Zdd = new Complex();
        Complex Ic = new Complex();
        Complex T5 = new Complex();
        Complex Zabr = new Complex();
        Complex W = new Complex();
        Complex Y1 = new Complex();
        Complex T1 = new Complex();
        Complex Zqq = new Complex();
        Complex Zqd = new Complex();
        Complex Zdq = new Complex();
        Complex Y2 = new Complex();
        Complex T2 = new Complex();
        Complex U1 = new Complex();
        Complex T3 = new Complex();
        Complex T4 = new Complex();
        Complex T6 = new Complex();
        Complex T7 = new Complex();
        Complex T8 = new Complex();
        Complex T9 = new Complex();
        Complex Iq = new Complex();
        Complex Id = new Complex();
        Complex T14 = new Complex();
        Complex k = new Complex();
        Complex T10 = new Complex();
        Complex T11 = new Complex();
        Complex T12 = new Complex();
        Complex T13 = new Complex();
        Complex Apr1 = new Complex();
        Complex wkdl = new Complex();
        Complex wkql = new Complex();
        Complex K1 = new Complex();
        Complex k2 = new Complex();
        Complex Aabr1 = new Complex();
        Complex T15 = new Complex();
        Complex T16 = new Complex();
        Complex T17 = new Complex();
        Complex T18 = new Complex();
        Complex T19 = new Complex();
        Complex T20 = new Complex();
        Complex K3 = new Complex();

        double Qh = Zs / (2 * p1);

```

```

double alfs1 = 2 * pi * p1 / Zs;
double alfr1 = 2 * pi * p1 / Zr;
double qd = Zs / (2 * p1 * m1);
double Kd = Math.Sin(qd * alfs1 / 2) *
Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
double Kq = Kd;

double Tetd1 = -2.618, Tetq1 = -1.047;
double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
* Math.Sin(alfr1 / 2) / alfr1;
double Tay = (pi * Di) / (2 * p1), ts =
pi * Di / Zs;
double Dr = Di - 2 * del, tr = pi * Dr /
Zr, lr = li;
double bhs = 0.002, hhs = 0.0005;
double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
double hps = (Da - Di - 2 * has) / 2;
double h12s = hps - hhs - d1s / 2;
double bhr = 0.001, hhr = 0.0007;
double d1r = (pi * (Dr - 2 * hhr) - bzs *
Zr) / (Zr + pi);
double hpr = d1r + hhr, Spr = pi *
Math.Pow(d1r, 2) / 4;
double Sk1 = Spr * 0.625 / Math.Sin(alfr1
/ 2), bk1 = 0.009;
double ak1 = Sk1 / bk1, Dk1 = Dr - bk1;
double bck = 1, Gamck = 2 * bck * ts /
Dr;
double kck = 2 * Math.Sin(p1 * Gamck / 2)
/ (p1 * Gamck);
double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);
double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;
double K0 = 1.55, B0 = 1, Ysr = 11.741 /
3.0;
double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = K0 * TayY + 0.02 * B0;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hk1 = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hk1 - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;
double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmu1), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dk1 / (Zr * ak1 * bk1);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +

```

```

x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);
double x11o = x11 / (x11 + x21h), x21ho =
x21h / (x11 + x21h);
double r11o = r11 / (x11 + x21h), r21ho =
r21h / (x11 + x21h);
double a = 2 * r21h * (r11o * (r11o +
r21ho) - x11o * (1 + x21ho));
double b = 2 * r11o * (1 + x11o) * x21h;
double c = 2 * r21h * (r11o * (1 + x21ho)
+ x11o * (r11o + r21ho));
double d = 2 * x21h * (x11o -
Math.Pow(r11o, 2));
double a1 = r21ho * (r21ho + 2 * r11o),
b1 = Math.Pow(r11o, 2) - 1;
double c1 = 2 * r21ho, d1 = 2 * r11o;
double e = 2 * r21h * (Math.Pow(r11o, 2)
- Math.Pow(x11o, 2));
double f = 4 * x11o * r11o * r21h;
double m = r21h * (r21h * (Math.Pow(r11o,
2) - Math.Pow(x11o, 2)) - 4 * r11o * x11o * x21h);
double n = Math.Pow(x21h, 2) *
(Math.Pow(r11o, 2) - Math.Pow(x11o, 2));
double p = 2 * ((Math.Pow(r11o, 2) -
Math.Pow(x11o, 2)) * x21h + r11o * x11o * r21h) *
r21h;
double Q = 2 * r11o * x11o *
Math.Pow(x21h, 2);
double Ah = (rd * rq - xd * xq) * a1 -
(rd * xq + rq * xd) * c1 + Math.Pow(wd * Kd, 2) * (rq
* a - xq * c) +
Math.Pow(wq * Kq, 2) * (rd * a - xd *
c) + 4 * Math.Pow(wd * Kd * wq * Kq, 2) * m;
double Bh = (rd * rq - xd * xq) * b1 -
(rd * xq + rq * xd) * d1 + Math.Pow(wd * Kd, 2) * (xq
* d - rq * b) +
Math.Pow(wq * Kq, 2) * (xd * d - rd *
b) - 4 * Math.Pow(wd * Kd * wq * Kq, 2) * n;
double Ch = (rd * rq - xd * xq) * c1 +
(rd * xq + rq * xd) * a1 + Math.Pow(wd * Kd, 2) * (rq
* c + xq * a) +
Math.Pow(wq * Kq, 2) * (rd * c + xd *
a) + 4 * Math.Pow(wd * Kd * wq * Kq, 2) * p;
double Dh = (rd * rq - xd * xq) * d1 +
(rd * xq + rq * xd) * b1 - Math.Pow(wd * Kd, 2) * (rq
* d + xq * b) -
Math.Pow(wq * Kq, 2) * (rd * d + xd *
b) - 4 * Math.Pow(wd * Kd * wq * Kq, 2) * Q;
double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
double AA = rd * a1 - xd * c1 +
Math.Pow(wd * Kd, 2) * a + Ah * gc - Ch * bc;
double BB = rd * b1 - xd * d1 -
Math.Pow(wd * Kd, 2) * b + Bh * gc - Dh * bc;
double CC = rd * c1 + xd * a1 +
Math.Pow(wd * Kd, 2) * c + Ch * gc + Ah * bc;
double DD = rd * d1 + xd * b1 -
Math.Pow(wd * Kd, 2) * d + Dh * gc + Bh * bc;

double Aq = rq * a1 - xq * c1 +
Math.Pow(wq * Kq, 2) * a;
double Bq = rq * b1 - xq * d1 -
Math.Pow(wq * Kq, 2) * b;
double Cq = rq * c1 + xq * a1 +
Math.Pow(wd * Kd, 2) * c; //ошибка в дипломе
(наверное, должно быть wq и Kq)
double Dq = rq * d1 + xq * b1 -
Math.Pow(wq * Kq, 2) * d;
double Eq = -wq * Kq * wd * Kd * f, Fq =
wq * Kq * wd * Kd * e;

double Ad = rd * a1 - xd * c1 +
Math.Pow(wd * Kd, 2) * a;
double Bd = rd * b1 - xd * d1 -
Math.Pow(wd * Kd, 2) * b;

```

```

        double Cd = rd * c1 + xd * a1 +
Math.Pow(wd * Kd, 2) * c;
        double Ddd = rd * d1 + xd * b1 -
Math.Pow(wd * Kd, 2) * d;
        double Ed = wd * Kd * wq * Kq * f, Fd = -
wd * Kd * wq * Kq * e;

        double Adq = Aq + Bq + Ad + Bd, Bdq = Cq
+ Dq + Cd + Ddd;
        double Ip = U * Math.Sqrt((Math.Pow(1 +
Adq * gc - Bdq * bc, 2) + Math.Pow(Adq * bc + Bdq *
gc, 2)) / (Math.Pow(AA + BB, 2) + Math.Pow(CC + DD,
2)));

        Spr = 0.0885;
        double Sabr = 2 - 0.0885;
        X.Re = r11;
        X.Im = x11;
        Y.Re = r21h;
        Y.Im = Spr * x21h;
        V.Re = Spr * r11 + r21h;
        V.Im = Spr * (x11 + x21h);
        Y1.Re = r21h;
        Y1.Im = Sabr * x21h;
        W.Re = Sabr * r11 + r21h;
        W.Im = Sabr * (x11 + x21h);
        T = X * Y;
        Zpr = T / V;
        T1 = X * Y1;
        Zabr = T1 / W;
        Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
        Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
        Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
        Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
        Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);
        Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);
        Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
Zabr.Im);
        Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
Zabr.Re);

        Y2.Re = gc;
        Y2.Im = bc;
        U1.Re = 220;
        U1.Im = 0;
        k.Re = 1;
        k.Im = 0;
        T2 = Zdd - Zqd;
        T3 = T2 * Y2;
        T4 = T3 * U1;
        T5 = Zdd * Zqq;
        T6 = Zdq * Zqd;
        T7 = T5 - T6;
        T8 = T7 * Y2;
        T9 = Zdd + T8;
        Iq = T4 / T9;
        T10 = Zqq - Zdq;
        T11 = T10 * Y2;
        T12 = T11 + k;
        T13 = T12 * U1;
        T14 = Zdd + T8;
        Id = T13 / T14;
        Ic = Id + Iq;
        double Ic1 = Complex.Abs(Ic);

        return Ip / Ic1;
}

public static double Gff(Vector<double> x) //
Коэффициент заполнения паза
{

```

```

        const double pi = Math.PI;

        double Di = x[0]; //диаметр расточки
статора
        double bzs = x[2]; // ширина зубца
статора
        double has = x[4]; //высота стенки
статора
        double Da = x[6]; // наружный диаметр
листа статора

        double Zs = 24, deliz = 0.00019, Up1 =
112;

        double hhs = 0.0005;
        double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
        double b2s = pi * (Da - 2 * has) / Zs -
bzs;
        double hps = (Da - Di - 2 * has) / 2;
        double h12s = hps - hhs - d1s / 2;
        double Sps = (pi * Math.Pow(d1s, 2)) / 8
+ h12s * (d1s + b2s) / 2;
        double Siz = 1.1 * deliz * (pi * d1s / 2
+ 2 * h12s + b2s);
        double diz = 0.00071;
        double kzz = Up1 * Math.Pow(diz, 2) /
(Sps - Siz);

        return kzz;
}

public static double Cosfi(Vector<double> x)
// Коэффициент мощности
{
        const double pi = Math.PI;

        double Di = x[0]; //диаметр расточки
статора
        double li = x[1]; //длина пакета статора
        double bzs = x[2]; // ширина зубца
статора
        double bzs = x[3]; // ширина зубца ротора
        double has = x[4]; //высота стенки
статора
        double Da = x[6]; // наружный диаметр
листа статора

        double Pn = 250, p1 = 2, f1 = 50, m1 = 2,
Cp = 10, Zs = 24, Zr = 17, w2k2 = 0.5, dg1 = 0.00063,
dg2 = 0.0005, ro75 = 0.000001 / 46.0;
        double del = 0.00025, deliz = 0.00019,
kpd1 = 0.6, roh = 0.000001 / 22.5, mu0 = 4 * pi *
0.0000001, kmu1 = 1.48;
        double lampr = 1.39, lamlr = 0.2, Up1 =
112, Up2 = 162;

        Complex X = new Complex();
        Complex Y = new Complex();
        Complex Zpr = new Complex();
        Complex V = new Complex();
        Complex T = new Complex();
        Complex Zdd = new Complex();
        Complex Ic = new Complex();
        Complex T5 = new Complex();
        Complex Zabr = new Complex();
        Complex W = new Complex();
        Complex Y1 = new Complex();
        Complex T1 = new Complex();
        Complex Zqq = new Complex();
        Complex Zqd = new Complex();
        Complex Zdq = new Complex();
        Complex Y2 = new Complex();
        Complex T2 = new Complex();

```

```

Complex U1 = new Complex();
Complex T3 = new Complex();
Complex T4 = new Complex();
Complex T6 = new Complex();
Complex T7 = new Complex();
Complex T8 = new Complex();
Complex T9 = new Complex();
Complex Iq = new Complex();
Complex Id = new Complex();
Complex T14 = new Complex();
Complex k = new Complex();
Complex T10 = new Complex();
Complex T11 = new Complex();
Complex T12 = new Complex();
Complex T13 = new Complex();
Complex Apr1 = new Complex();
Complex wkdl = new Complex();
Complex wkql = new Complex();
Complex K1 = new Complex();
Complex k2 = new Complex();
Complex Aabr1 = new Complex();
Complex T15 = new Complex();
Complex T16 = new Complex();
Complex T17 = new Complex();
Complex T18 = new Complex();
Complex T19 = new Complex();
Complex T20 = new Complex();
Complex K3 = new Complex();
double Qh = Zs / (2 * p1);
double alfs1 = 2 * pi * p1 / Zs;
double alfr1 = 2 * pi * p1 / Zr;
double qd = Zs / (2 * p1 * m1);
double Kd = Math.Sin(qd * alfs1 / 2) *
Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
double Kq = Kd;
double Tetd1 = -2.618, Tetq1 = -1.047;
double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
* Math.Sin(alfr1 / 2) / alfr1;
double kt = 6;
double n1 = 60 * f1 / p1;
double Pmex = kt * Math.Pow(n1 / 1000, 2)
* Math.Pow(Da * 10, 3);
double Pdab = 0.005 * Pn / kpd1, Prach =
Pn + Pmex + Pdab, Mt = 9.55 * (Pmex + Pdab) / n1;
double Tay = (pi * Di) / (2 * p1), ts =
pi * Di / Zs;
double Dr = Di - 2 * del, tr = pi * Dr /
Zr, lr = li;
double bhs = 0.002, hhs = 0.0005;
double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
double b2s = pi * (Da - 2 * has) / Zs -
bzs;
double hps = (Da - Di - 2 * has) / 2;
double h12s = hps - hhs - d1s / 2;
double Sps = (pi * Math.Pow(d1s, 2)) / 8
+ h12s * (d1s + b2s) / 2;
double Siz = 1.1 * deliz * (pi * d1s / 2
+ 2 * h12s + b2s);
double Spsh = Sps - Siz;
double bhr = 0.001, hhr = 0.0007;
double d1r = (pi * (Dr - 2 * hhr) - b2r *
Zr) / (Zr + pi);
double hpr = d1r + hhr, Spr = pi *
Math.Pow(d1r, 2) / 4;
double Sk1 = Spr * 0.625 / Math.Sin(alfr1
/ 2), bk1 = 0.009;
double ak1 = Sk1 / bk1, Dk1 = Dr - bk1;
double bck = 1, Gamck = 2 * bck * ts /
Dr;
double kck = 2 * Math.Sin(p1 * Gamck / 2)
/ (p1 * Gamck);
double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);

double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;
double KO = 1.55, BO = 1, Ysr = 11.741 /
3.0;
double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = KO * TayY + 0.02 * BO;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hk1 = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hk1 - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;
double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmu1), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dk1 / (Zr * ak1 * bk1);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +
x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);
double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
Spr = 0.0885;
double Sabr = 2 - 0.0885;
X.Re = r11;
X.Im = x11;
Y.Re = r21h;
Y.Im = Spr * x21h;
V.Re = Spr * r11 + r21h;
V.Im = Spr * (x11 + x21h);
Y1.Re = r21h;
Y1.Im = Sabr * x21h;
W.Re = Sabr * r11 + r21h;
W.Im = Sabr * (x11 + x21h);
T = X * Y;
Zpr = T / V;
T1 = X * Y1;
Zabr = T1 / W;
Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);
Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);

```

```

Zabr.Im);
Zabr.Re);
    Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
    Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
    Y2.Re = gc;
    Y2.Im = bc;
    U1.Re = 220;
    U1.Im = 0;
    k.Re = 1;
    k.Im = 0;
    T2 = Zdd - Zqd;
    T3 = T2 * Y2;
    T4 = T3 * U1;
    T5 = Zdd * Zqq;
    T6 = Zdq * Zqd;
    T7 = T5 - T6;
    T8 = T7 * Y2;
    T9 = Zdd + T8;
    Iq = T4 / T9;
    T10 = Zqq - Zdq;
    T11 = T10 * Y2;
    T12 = T11 + k;
    T13 = T12 * U1;
    T14 = Zdd + T8;
    Id = T13 / T14;
    Ic = Id + Iq;
    double Ic1 = Complex.Abs(Ic);
    double COSfi = Ic.Re / Ic1;

    return COSfi;
}

public static double Eff(Vector<double> x) //
Кoeffициент полезного действия
{
    const double pi = Math.PI;

    double Di = x[0]; //диаметр расточки
    статора
    double li = x[1]; //длина пакета статора
    double bzs = x[2]; // ширина зубца
    статора
    double bzs = x[3]; // ширина зубца ротора
    double has = x[4]; //высота стенки
    статора
    double har = x[5]; //высота стенки ротора
    double Da = x[6]; // наружный диаметр
    листа статора

    double Pn = 250, U = 220, p1 = 2, f1 =
    50, m1 = 2, Cp = 10, Zs = 24, Zr = 17, dv = 0.02,
    w2k2 = 0.5, dg1 = 0.00063, dg2 = 0.0005, ro75 =
    0.000001 / 46.0;
    double del = 0.00025, deliz = 0.00019,
    kpd1 = 0.6, roh = 0.000001 / 22.5, mu0 = 4 * pi *
    0.0000001, kmu1 = 1.48, kc = 0.95, Kts = 0.9, Ktr =
    1.3;
    double lampr = 1.39, lamlr = 0.2, Up1 =
    112, Up2 = 162;

    Complex X = new Complex();
    Complex Y = new Complex();
    Complex Zpr = new Complex();
    Complex V = new Complex();
    Complex T = new Complex();
    Complex Zdd = new Complex();
    Complex Ic = new Complex();
    Complex T5 = new Complex();
    Complex Zabr = new Complex();
    Complex W = new Complex();
    Complex Y1 = new Complex();
    Complex T1 = new Complex();
    Complex Zqq = new Complex();
    Complex Zqd = new Complex();
    Complex Zdq = new Complex();

    Complex Y2 = new Complex();
    Complex T2 = new Complex();
    Complex U1 = new Complex();
    Complex T3 = new Complex();
    Complex T4 = new Complex();
    Complex T6 = new Complex();
    Complex T7 = new Complex();
    Complex T8 = new Complex();
    Complex T9 = new Complex();
    Complex Iq = new Complex();
    Complex Id = new Complex();
    Complex T14 = new Complex();
    Complex k = new Complex();
    Complex T10 = new Complex();
    Complex T11 = new Complex();
    Complex T12 = new Complex();
    Complex T13 = new Complex();
    Complex Apr1 = new Complex();
    Complex wkd1 = new Complex();
    Complex wkq1 = new Complex();
    Complex K1 = new Complex();
    Complex k2 = new Complex();
    Complex Aabr1 = new Complex();
    Complex T15 = new Complex();
    Complex T16 = new Complex();
    Complex T17 = new Complex();
    Complex T18 = new Complex();
    Complex T19 = new Complex();
    Complex T20 = new Complex();
    Complex K3 = new Complex();
    double Qh = Zs / (2 * p1);
    double alfs1 = 2 * pi * p1 / Zs;
    double alfr1 = 2 * pi * p1 / Zr;
    double qd = Zs / (2 * p1 * m1);
    double Kd = Math.Sin(qd * alfs1 / 2) *
    Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
    double Kq = Kd;
    double Tetd1 = -2.618, Tetq1 = -1.047;
    double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
    * Math.Sin(alfr1 / 2) / alfr1;
    double kt = 6;
    double n1 = 60 * f1 / p1;
    double Pmex = kt * Math.Pow(n1 / 1000, 2)
    * Math.Pow(Da * 10, 3);
    double Pdab = 0.005 * Pn / kpd1, Prach =
    Pn + Pmex + Pdab, Mt = 9.55 * (Pmex + Pdab) / n1;
    double Tay = (pi * Di) / (2 * p1), ts =
    pi * Di / Zs;
    double Dr = Di - 2 * del, tr = pi * Dr /
    Zr, lr = li;
    double bhs = 0.002, hhs = 0.0005;
    double d1s = (pi * (Di + 2 * hhs) - bzs *
    Zs) / (Zs - pi);
    double b2s = pi * (Da - 2 * has) / Zs -
    bzs;
    double hps = (Da - Di - 2 * has) / 2;
    double h12s = hps - hhs - d1s / 2;
    double Sps = (pi * Math.Pow(d1s, 2)) / 8
    + h12s * (d1s + b2s) / 2;
    double Siz = 1.1 * deliz * (pi * d1s / 2
    + 2 * h12s + b2s);
    double Spsh = Sps - Siz;
    double bhr = 0.001, hhr = 0.0007;
    double d1r = (pi * (Dr - 2 * hhr) - bzs *
    Zr) / (Zr + pi);
    double hpr = d1r + hhr, Spr = pi *
    Math.Pow(d1r, 2) / 4;
    double Sk1 = Spr * 0.625 / Math.Sin(alfr1
    / 2), bkl = 0.009;
    double akl = Sk1 / bkl, Dkl = Dr - bkl;
    double bck = 1, Gamck = 2 * bck * ts /
    Dr;
    double kck = 2 * Math.Sin(p1 * Gamck / 2)
    / (p1 * Gamck);

```

```

double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);
double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;
double KO = 1.55, B0 = 1, Ysr = 11.741 /
3.0;
double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = KO * TayY + 0.02 * B0;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hk1 = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hk1 - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;
double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmu1), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dk1 / (Zr * ak1 * bk1);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +
x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);
double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
Spr = 0.0885;
double Sabr = 2 - 0.0885;
X.Re = r11;
X.Im = x11;
Y.Re = r21h;
Y.Im = Spr * x21h;
V.Re = Spr * r11 + r21h;
V.Im = Spr * (x11 + x21h);
Y1.Re = r21h;
Y1.Im = Sabr * x21h;
W.Re = Sabr * r11 + r21h;
W.Im = Sabr * (x11 + x21h);
T = X * Y;
Zpr = T / V;
T1 = X * Y1;
Zabr = T1 / W;
Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);

```

```

Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);
Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
Zabr.Im);
Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
Zabr.Re);
Y2.Re = gc;
Y2.Im = bc;
U1.Re = 220;
U1.Im = 0;
k.Re = 1;
k.Im = 0;
T2 = Zdd - Zqd;
T3 = T2 * Y2;
T4 = T3 * U1;
T5 = Zdd * Zqq;
T6 = Zdq * Zqd;
T7 = T5 - T6;
T8 = T7 * Y2;
T9 = Zdd + T8;
Iq = T4 / T9;
T10 = Zqq - Zdq;
T11 = T10 * Y2;
T12 = T11 + k;
T13 = T12 * U1;
T14 = Zdd + T8;
Id = T13 / T14;
Ic = Id + Iq;
double Rc = 0.00001, Id1 =
Complex.Abs(Id), Iq1 = Complex.Abs(Iq), Ic1 =
Complex.Abs(Ic);
wkd1.Re = 672 * 0.9106;
wkd1.Im = 0;
wkq1.Re = 972 * 0.9106;
wkq1.Im = 0;
K1.Re = 0;
K1.Im = -1;
k2.Re = 0;
k2.Im = 1;
T15 = Id * wkd1;
T16 = Iq * wkq1;
T17 = T16 * K1;
Apr1 = T15 + T17;
double Apr11Apr11 =
Math.Pow(Complex.Abs(Apr1), 2);
T18 = T16 * k2;
Aabr1 = T15 + T18;
double Aabr11Aabr11 =
Math.Pow(Complex.Abs(Aabr1), 2);
double Alfd = 0.6782, kb = 1.117;
T19 = Zpr * Apr1;
T20 = Zabr * Aabr1;
K3 = T19 + T20;
double K33 = Complex.Abs(K3);
double Fr = K33 / (4 * kb * f1);
double K44 = Complex.Abs(T19);
double Fpr = K44 / (4 * kb * f1);
double K55 = Complex.Abs(T20);
double Fabr = K55 / (4 * kb * f1);
double Bdel = Fr / (Alfd * Tay * li),
Bdelpr = Fpr / (Alfd * Tay * li), Bdelabr = Fabr /
(Alfd * Tay * li);
double Bzsh = Bdel * (ts / (bzs * kc)),
Bzshpr = Bdelpr * (ts / (bzs * kc)), Bzshabr =
Bdelabr * (ts / (bzs * kc));
double Bzrh = Bdel * (tr / (bzh * kc)),
Bzrhpr = Bdelpr * (tr / (bzh * kc)), Bzrhabr =
Bdelabr * (tr / (bzh * kc));
double Bas = Fr / (2 * li * has * kc),
Baspr = Fpr / (2 * li * has * kc), Basabr = Fabr / (2
* li * has * kc);
double Bar = Fr / (2 * lr * har * kc),
Barpr = Fpr / (2 * lr * har * kc), Barabr = Fabr / (2
* lr * har * kc);

```



```

Complex T6 = new Complex();
Complex T7 = new Complex();
Complex T8 = new Complex();
Complex T9 = new Complex();
Complex Iq = new Complex();
Complex Id = new Complex();
Complex T14 = new Complex();
Complex k = new Complex();
Complex T10 = new Complex();
Complex T11 = new Complex();
Complex T12 = new Complex();
Complex T13 = new Complex();
Complex Apr1 = new Complex();
Complex wkdl = new Complex();
Complex wkql = new Complex();
Complex K1 = new Complex();
Complex k2 = new Complex();
Complex Aabr1 = new Complex();
Complex T15 = new Complex();
Complex T16 = new Complex();
Complex T17 = new Complex();
Complex T18 = new Complex();
Complex T19 = new Complex();
Complex T20 = new Complex();
Complex K3 = new Complex();

double Qh = Zs / (2 * p1);
double alfs1 = 2 * pi * p1 / Zs;
double alfr1 = 2 * pi * p1 / Zr;
double qd = Zs / (2 * p1 * m1);
double Kd = Math.Sin(qd * alfs1 / 2) *
Math.Sin(pi / 2) / (qd * Math.Sin(alfs1 / 2));
double Kq = Kd;
double Tetd1 = -2.618, Tetq1 = -1.047;
double Tet1 = pi + Tetd1 - Tetq1, Kp1 = 2
* Math.Sin(alfr1 / 2) / alfr1;
double kt = 6;
double n1 = 60 * f1 / p1;
double Pmex = kt * Math.Pow(n1 / 1000, 2)
* Math.Pow(Da * 10, 3);
double Tay = (pi * Di) / (2 * p1), ts =
pi * Di / Zs;
double Dr = Di - 2 * del, tr = pi * Dr /
Zr, lr = li;
double bhs = 0.002, hhs = 0.0005;
double d1s = (pi * (Di + 2 * hhs) - bzs *
Zs) / (Zs - pi);
double b2s = pi * (Da - 2 * has) / Zs -
bzs;
double hps = (Da - Di - 2 * has) / 2;
double h12s = hps - hhs - d1s / 2;
double Sps = (pi * Math.Pow(d1s, 2)) / 8
+ h12s * (d1s + b2s) / 2;
double Siz = 1.1 * deliz * (pi * d1s / 2
+ 2 * h12s + b2s);
double bhr = 0.001, hhr = 0.0007;
double d1r = (pi * (Dr - 2 * hhr) - bzs *
Zr) / (Zr + pi);
double hpr = d1r + hhr, Spr = pi *
Math.Pow(d1r, 2) / 4;
double Sk1 = Spr * 0.625 / Math.Sin(alfr1
/ 2), bkl = 0.009;
double ak1 = Sk1 / bkl, Dk1 = Dr - bkl;
double bck = 1, Gamck = 2 * bck * ts /
Dr;
double kck = 2 * Math.Sin(p1 * Gamck / 2)
/ (p1 * Gamck);
double Kdelr = (5 + bhr / del) / (5 + bhr
/ del * (tr - bhr) / tr);
double Kdels = (5 + bhs / del) / (5 + bhs
/ del * (ts - bhs) / ts);
double delh = del * Kdelr * Kdels;
double KO = 1.55, B0 = 1, Ysr = 11.741 /
3.0;

double Tsr = pi * (Di + hps) / Zs, TayY =
Tsr * Ysr;
double L1 = KO * TayY + 0.02 * B0;
double Ld = 2 * li + 2 * L1, Lv = 0.235 *
p1 * Tay;
double wd = Up1 * p1 * qd, wq = Up2 * p1
* qd;
double rd = ro75 * Ld * wd / (pi *
Math.Pow(dg1, 2) / 4), rq = ro75 * Ld * wq / (pi *
Math.Pow(dg2, 2) / 4);
double Hk1 = 0.00025, Kbeta = 1, Kbeta1 =
1, ksi = 1.35;
double h1 = h12s - 2 * deliz, h2 = 2 *
deliz + Hk1 - 0.5 * d1s;
double Lamp = h1 * Kbeta / (3 * d1s) +
(0.785 - bhs / (2 * d1s) + h2 / d1s + hhs / bhs) *
Kbeta1;
double Kdel = Kdels * Kdelr;
double Lamd = ts * ksi / (11.9 * del *
Kdel), Lam1 = 0.27 * qd / li * (L1 - 0.64 * Tay);
double SLam = Lamp + Lamd + Lam1;
double xd = 0.158 * f1 * Math.Pow(wd /
100.0, 2) * li / (p1 * qd) * SLam;
double xq = 0.158 * f1 * Math.Pow(wq /
100.0, 2) * li / (p1 * qd) * SLam;
double x11 = (4 * f1 * Tay * li * mu0) /
(pi * p1 * delh * kmul), Spr1 = (pi * Math.Pow(d1r,
2)) / 4;
double rct = roh * lr / Spr1, r1 = roh *
pi * Dk1 / (Zr * ak1 * bkl);
double r21h = (rct + r1 / (2 *
Math.Pow(Math.Sin(alfr1 / 2), 2))) / (Zr *
Math.Pow(w2k2 * kck, 2));
double r11 = 0.182 / 10000.0;
double x21h = (2 * pi * f1 * mu0 * lr) /
(Zr * Math.Pow(w2k2 * kck, 2)) * (lampr + lam1r / (4
* Math.Pow(Math.Sin(alfr1 / 2), 2))) +
x11 / 2 * (1 / (Math.Pow(Kp1, 2) *
Math.Pow(kck, 2)) - 1);
double bc = 2 * pi * f1 * Cp * 0.000001;
double gc = bc * Math.Tan(0.01);
Spr = 0.0885;
double Sabr = 2 - 0.0885;
X.Re = r11;
X.Im = x11;
Y.Re = r21h;
Y.Im = Spr * x21h;
V.Re = Spr * r11 + r21h;
V.Im = Spr * (x11 + x21h);
Y1.Re = r21h;
Y1.Im = Sabr * x21h;
W.Re = Sabr * r11 + r21h;
W.Im = Sabr * (x11 + x21h);
T = X * Y;
Zpr = T / V;
T1 = X * Y1;
Zabr = T1 / W;
Zdd.Re = rd + Math.Pow(wd * Kd, 2) *
(Zpr.Re + Zabr.Re);
Zdd.Im = xd + Math.Pow(wd * Kd, 2) *
(Zpr.Im + Zabr.Im);
Zqq.Re = rq + Math.Pow(wq * Kq, 2) *
(Zpr.Re + Zabr.Re);
Zqq.Im = xq + Math.Pow(wq * Kq, 2) *
(Zpr.Im + Zabr.Im);
Zqd.Re = wd * Kd * wq * Kq * (-Zpr.Im +
Zabr.Im);
Zqd.Im = wd * Kd * wq * Kq * (Zpr.Re -
Zabr.Re);
Zdq.Re = wd * Kd * wq * Kq * (Zpr.Im -
Zabr.Im);
Zdq.Im = wd * Kd * wq * Kq * (-Zpr.Re +
Zabr.Re);
Y2.Re = gc;
Y2.Im = bc;

```

```

U1.Re = 220;
U1.Im = 0;
k.Re = 1;
k.Im = 0;
T2 = Zdd - Zqd;
T3 = T2 * Y2;
T4 = T3 * U1;
T5 = Zdd * Zqq;
T6 = Zdq * Zqd;
T7 = T5 - T6;
T8 = T7 * Y2;
T9 = Zdd + T8;
Iq = T4 / T9;
T10 = Zqq - Zdq;
T11 = T10 * Y2;
T12 = T11 + k;
T13 = T12 * U1;
T14 = Zdd + T8;
Id = T13 / T14;
Ic = Id + Iq;
double Rc = 0.00001, Id1 =
Complex.Abs(Id), Iq1 = Complex.Abs(Iq), Ic1 =
Complex.Abs(Ic);
wkd1.Re = 672 * 0.9106;
wkd1.Im = 0;
wkq1.Re = 972 * 0.9106;
wkq1.Im = 0;
K1.Re = 0;
K1.Im = -1;
k2.Re = 0;
k2.Im = 1;
T15 = Id * wkd1;
T16 = Iq * wkq1;
T17 = T16 * K1;
Apr1 = T15 + T17;
double Apr11Apr11 =
Math.Pow(Complex.Abs(Apr1), 2);
T18 = T16 * k2;
Aabr1 = T15 + T18;
double Aabr11Aabr11 =
Math.Pow(Complex.Abs(Aabr1), 2);
double Alfd = 0.6782, kb = 1.117;
T19 = Zpr * Apr1;
T20 = Zabr * Aabr1;
K3 = T19 + T20;
double K33 = Complex.Abs(K3);
double Fr = K33 / (4 * kb * f1);
double K44 = Complex.Abs(T19);
double Fpr = K44 / (4 * kb * f1);
double K55 = Complex.Abs(T20);
double Fabr = K55 / (4 * kb * f1);
double Bdel = Fr / (Alfd * Tay * li),
Bdelpr = Fpr / (Alfd * Tay * li), Bdelabr = Fabr /
(Alfd * Tay * li);
double Bzsh = Bdel * (ts / (bzs * kc)),
Bzshpr = Bdelpr * (ts / (bzs * kc)), Bzshabr =
Bdelabr * (ts / (bzs * kc));
double Bzrh = Bdel * (tr / (bzs * kc)),
Bzrhpr = Bdelpr * (tr / (bzs * kc)), Bzrhabr =
Bdelabr * (tr / (bzs * kc));
double Bas = Fr / (2 * li * has * kc),
Baspr = Fpr / (2 * li * has * kc), Basabr = Fabr / (2
* li * has * kc);
double Bar = Fr / (2 * lr * har * kc),
Barpr = Fpr / (2 * lr * har * kc), Barabr = Fabr / (2
* lr * har * kc);
double p_10_50 = 3.3, Gas = 7800.0 * pi *
(Da - has) * has * li * kc;
double Paspr = 1.6 * p_10_50 *
Math.Pow(Baspr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
double Pasabr = 1.6 * p_10_50 *
Math.Pow(Basabr, 2) * Gas * Math.Pow(f1 / 50, 1.3);
double Gzs = 7800.0 * Zs * bzs * hps * li
* kc;

double Pzspr = 1.8 * p_10_50 *
Math.Pow(Bzshpr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
double Pzsabr = 1.8 * p_10_50 *
Math.Pow(Bzshabr, 2) * Gzs * Math.Pow(f1 / 50, 1.3) *
Kts;
double Gar = 7800.0 * pi * (dv + har) *
har * li * kc;
double Parpr = 1.6 * p_10_50 *
Math.Pow(Barpr, 2) * Gar * Math.Pow(f1 * Spr / 50,
1.3);
double Parabr = 1.6 * p_10_50 *
Math.Pow(Barabr, 2) * Gar * Math.Pow(f1 * Sabr / 50,
1.3);
double Gzr = 7800.0 * Zr * bzs * hpr * li
* kc;
double Pzrpr = 1.8 * p_10_50 *
Math.Pow(Bzrhpr, 2) * Gzr * Math.Pow(f1 * Spr / 50,
1.3) * Ktr;
double Pzrabr = 1.8 * p_10_50 *
Math.Pow(Bzrhabr, 2) * Gzr * Math.Pow(f1 * Sabr / 50,
1.3) * Ktr;
double Pctabr = Pasabr + Pzsabr + Parabr
+ Pzrabr, Pctpr = Paspr + Pzspr + Parpr + Pzrpr;
double r11pr1 = 0.1820 * 0.0001, r11abr1
= 0.334 * 0.0001;
double Zmupr = Math.Sqrt(Math.Pow(r11pr1,
2) + Math.Pow(x11, 2));
double r11pr = Pctpr * Math.Pow(Zmupr, 2)
/ Math.Pow(K44, 2);
double Zmuabr =
Math.Sqrt(Math.Pow(r11abr1, 2) + Math.Pow(x11, 2));
double r11abr = Pctabr * Math.Pow(Zmuabr,
2) / Math.Pow(K55, 2);
double Apr11 = Math.Sqrt(Apr11Apr11),
Aabr11 = Math.Sqrt(Aabr11Aabr11);
double Ppsipr = Math.Pow(Apr11, 2) *
((Math.Pow(r11pr, 2) + Math.Pow(x11, 2)) * r21h *
Spr) / (Math.Pow(r11pr * Spr + r21h, 2) +
Math.Pow(Spr, 2) * Math.Pow(x11 + x21h, 2));
double Ppsiabr = Math.Pow(Aabr11, 2) *
((Math.Pow(r11abr, 2) + Math.Pow(x11, 2)) * r21h *
Sabr) / (Math.Pow(r11abr * Sabr + r21h, 2) +
Math.Pow(Sabr, 2) * Math.Pow(x11 + x21h, 2));
double Pmr = Ppsipr * Spr + Ppsiabr *
Sabr;
double Pms = Math.Pow(Id1, 2) * rd +
Math.Pow(Iq1, 2) * rq;
double kp = 0, ktl = 0.162, ktem = 0.143,
ktip = 0.28, diz = 0.00071, tetasr = 53, dellh =
0.00004, delhh = 0.0001, ktv = 0.0264, bps = 0.007;
double PP = 2 * hps + bps, kzz = Up1 *
Math.Pow(diz, 2) / (Sps - Siz);
double DEL1 = diz * (0.022 + 0.5 * (1 /
Math.Sqrt(kzz) - 1));
double kte = 0.165 * (1 + 0.0007 *
tetasr) * (1 - 32 * diz * (1 - 9.2 * kp + 5.2 *
Math.Pow(kp, 2)) +
8100.0 * Math.Pow(diz, 2)) * (2.11 *
Math.Pow(kzz, 1.5) - 0.32) * Math.Pow(ktl / 0.162,
1.3) * Math.Pow(ktem / 0.143, 0.25);
double Rps = (1 / (PP * li * Zs)) * ((bps
- 2 * deliz) * (dellh + DEL1) / ((diz + DEL1) * kte)
+ deliz / ktip + delhh / ktv);
double kv = 2.2 * 0.00001, blr = 0.0115;
double dek = 4 * bps * hps / (2 * (bps +
hps)), wl = 2.2 * 0.01 * Dr * n1;
double Ree = wl * dek / kv;
double Nu = 0.103 * Math.Pow(Ree, 0.66);
double alfnl = (Nu * ktv) / dek, Re1 =
wl * blr / kv;
double Nu1 = 0.456 * Math.Pow(Re1, 0.6);
double alfv1 = (Nu1 * ktv) / blr;
double S1 = (pi / 4) * L1 * Zs * (bps + 2
* hps);

```

```

double lg_kzat = 0.1 - 0.00323 * (Ll /
Lv) * Zs;
double kzat = Math.Pow(10, lg_kzat);
double Rkl = 2 / ((alfnl + alfv1) * S1 *
kzat);
double Rvl = hps * (del1h + DEL1) / (4 *
pi * (diz + DEL1) * kte * (Di + hps) * Lv);
double Rls = 0.5 * (Rkl + Rvl);
double ktm = 390, Sm = (pi *
Math.Pow(dg1, 2) / 4) * Up1;
double Rlp = 0.5 * (Ll + li) / (6 * ktm *
Sm * Zs);
double wv = 0, ktc = 40.6, Sc = pi * Da *
li;
double ak = 20 + 280 * Math.Pow(wv, 0.6);
double Ras = 1 / (ak * Sc);
Rc = 0.5 * has / (ktc * pi * (Da - has) *
li * kc);
double Rz = 0.5 * hps / (ktc * Zs * li *
bsz * kc);
double Rcs = Ras + Rc + Rz;
double llr = 0.017, nlr = 6, alr = 0.004;
double St = 2 * (2 * blr * llr * nlr +
0.785 * (Math.Pow(Dr, 2) - Math.Pow(dv, 2)));
double Rtr = 1 / (alfv1 * St);
double alfv = ktv / del;
double Rdel = 1 / (alfv * pi * Dr * lr);
double Ccte = Rps + Rcs;

double Bcte = Rdel + Rcs + Rtr;
double Acte = Rlp + Rps + Rcs + Rls;
double Pcs = Paspr + Pasabr + Pzspr +
Pzsabr;
double Pmp = Pms * li / (li + Ll);
double Pml = Pms * Ll / (li + Ll);
double Qlp = (Bcte * (Pmp * Ccte + Pcs *
Rcs - Pml * Rls) + Rcs * (Pmr * Rtr - Pcs * Rcs - Pmp
* Rcs)) / (Acte * Bcte - Math.Pow(Rcs, 2));
double Q1 = Pmp - Qlp, Q2 = Pml + Qlp;
double Qdel = (Pmr * Rtr - (Pcs + Q1) *
Rcs) / Bcte;
double Qc = Q1 + Pcs + Qdel, Qr = Pmr -
Qdel;
double Sp = Pmp + Pml + Pmr + Pcs + Pmex;
double Pv = 0.6 * 2.2 * 0.01 * n1 * Dr *
llr * (pi * (Dr - blr) - nlr * alr);
double Tetav = Sp / (2200.0 * Pv);
double Tetamp = Q1 * Rps + Qc * Rcs +
Tetav;
double Tetam1 = Q2 * Rls + Tetav;
double Tetasc1 = (Tetam1 * Ll + Tetamp *
li) / (Ll + li);

return Tetasc1;
}
}
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using MathNet.Numerics;
using MathNet.Numerics.LinearAlgebra;

```

```
namespace Диссертация.Calculation
```

```

{
    public class MathFunctions
    {
        public static Vector<double>
Gradient(Func<Vector<double>, double> f,
Vector<double> x, double h)
        {
            int n = x.Count;
            Vector<double> res =
Vector<double>.Build.Dense(n);
            for (int i = 0; i < n; i++)
            {
                Vector<double> x1 =
Vector<double>.Build.Dense(n);
                for (int j = 0; j < n; j++)
                {
                    if (i != j) x1[j] = x[j];
                    else x1[j] = x[j] + h;
                }
                Vector<double> x2 =
Vector<double>.Build.Dense(n);
                for (int j = 0; j < n; j++)
                {
                    if (i != j) x2[j] = x[j];
                    else x2[j] = x[j] - h;
                }
                res[i] = (f(x1) - f(x2)) / (2 * h);
            }
            return res;
        }

        public static double Cut(double x)
        {
            if (x >= 0) return 0;
            else return x;
        }
    }
}

```

```

}

public static double Square_Cut(double x)
{
    if (x >= 0) return 0;
    else return Math.Pow(x, 2);
}

public static bool
Good_X(List<Func<Vector<double>, double>> g,
Vector<double> x)
{
    int t = 0;
    int m = g.Count();
    for (int i = 0; i < m; i++)
    {
        if (g[i](x) < 0)
        {
            t++;
            break;
        }
    }
    if (t > 0) return false;
    else return true;
}

public static bool Positive_X(Vector<double>
x)
{
    int t = 0;
    int n = x.Count;
    for (int i = 0; i < n; i++)
    {
        if (x[i] <= 0)
        {
            t++;
            break;
        }
    }
    if (t > 0) return false;
    else return true;
}
}

```



```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using MathNet.Numerics;
using MathNet.Numerics.LinearAlgebra;

namespace Диссертация.Calculation
{
    public interface IStrategy
    {
        Vector<double> Optimize();
    }

    public class MethodFields
    {
        public Func<Vector<double>, double> f { get;
set; }
        public List<Func<Vector<double>, double>> g {
get; set; }
        public Vector<double> x { get; set; }
/*начальное значение*/
        public double h { get; set; }
        public int n { get; set; }
        public int m { get; set; }
        public Vector<double> R { get; set; }
        public Vector<double> sigma { get; set; }

        public MethodFields(Func<Vector<double>,
double> _f, List<Func<Vector<double>, double>> _g,
Vector<double> _x)
        {
            h = 0.00000001;
            x = _x;
            f = _f;
            g = new List<Func<Vector<double>,
double>>();
            foreach(Func<Vector<double>, double> func
in _g)
            {
                g.Add(func);
            }
            n = x.Count;
            m = g.Count;
        }

        public class Allowable_directions_Method :
MethodFields, IStrategy
        {
            public
Allowable_directions_Method(Func<Vector<double>,
double> _f, List<Func<Vector<double>, double>> _g,
Vector<double> _x) : base(_f, _g, _x)
            { }

            public bool
Good_X_for_allowable_directions_method(List<Func<Vect
or<double>, double>> g, Vector<double> x, int n,
double eps)
            {
                int t = 0;
                int m = g.Count();
                for (int i = 0; i < m; i++)
                {
                    if (g[i](x) < -eps) t++;
                }
                if (t > 0) return false;
                else return true;
            }

            public Vector<double> Optimize()
            {

```

```

                double eps1 = 0.00001; // для проверки
ограничений
                double eps2 = 0.00001; // для одномерной
                double eps3 = 0.0001; // окончательное
                int iter = 0;
                Vector<double> x_cur =
Vector<double>.Build.Dense(n);
                double alpha = 0;
                Vector<double> d =
Vector<double>.Build.Dense(n);
                Vector<double> x_new =
Vector<double>.Build.DenseOfVector(x);
                do
                {
                    iter++;
                    x.Clear();
                    x =
Vector<double>.Build.DenseOfVector(x_new);

                    // War 1
                    double[,] table = new double[2 + m +
2 * n, n + 2];

                    table[0, 0] = 0;
                    for (int i = 0; i < n; i++)
                    {
                        table[0, i + 1] =
MathFunctions.Gradient(f, x, h)[i];
                    }
                    table[0, n + 1] = 1;

                    for (int i = 0; i < m; i++)
                    {
                        table[i + 1, 0] = g[i](x);
                        for (int j = 0; j < n; j++)
                        {
                            table[i + 1, j + 1] = -
MathFunctions.Gradient(g[i], x, h)[j];
                        }
                        table[i + 1, n + 1] = 1;
                    }

                    for (int i = 0; i < n; i++)
                    {
                        table[m + 1 + i, 0] = 1;
                        table[m + 1 + i, i + 1] = 1;
                    }

                    for (int i = 0; i < n; i++)
                    {
                        table[m + 1 + n + i, 0] = 1;
                        table[m + 1 + n + i, i + 1] = -1;
                    }

                    for (int i = 0; i < n + 1; i++)
                    {
                        table[m + 1 + 2 * n, i] = 0;
                    }
                    table[m + 1 + 2 * n, n + 1] = -1;

                    double[,] table_new = new double[2 +
m + 2 * n, 2 * n + 3];

                    for (int i = 0; i < 2 + m + 2 * n;
i++)
                    {
                        table_new[i, 0] = table[i, 0];
                    }
                    for (int i = 0; i < 2 + m + 2 * n;
i++)
                    {
                        for (int j = 1; j < 2 * n + 3;
j++)
                        {

```

```

        if (j % 2 == 1) table_new[i,
j] = table[i, (j + 1) / 2];
        else table_new[i, j] = -
table[i, j / 2];
    }
}

double[] result = new double[2 * n +
2];

double[,] table_result;
Simplex S = new Simplex(table_new);
table_result = S.Calculate(result);

for (int i = 0; i < 2 * n; i += 2)
{
    d[i / 2] = result[i] - result[i +
1];
}

// Шаг 2
if ((result[2 * n] - result[2 * n +
1]) <= 0)
{
    break;
}
alpha = 0;
do
{
    alpha += 0.0001;
    x_cur.Clear();
    x_cur =
Vector<double>.Build.DenseOfVector(x + alpha * d);
    } while
(((!Good_X_for_allowable_directions_method(g, x_cur,
n, eps1))) && (alpha <= 0.1));

// Шаг 3
double l = 0, a = 0, b = 0, x1 = 0,
x2 = 0, fi = (1 + Math.Sqrt(5)) / 2, y1, y2;
// Метод золотого сечения
a = 0;
b = alpha;
x1 = b - (b - a) / fi;
x2 = a + (b - a) / fi;
do
{
    y1 = f(x + x1 * d);
    y2 = f(x + x2 * d);
    if (y1 >= y2)
    {
        a = x1;
        x1 = x2;
        x2 = a + (b - a) / fi;
    }
    else
    {
        b = x2;
        x2 = x1;
        x1 = b - (b - a) / fi;
    }
} while (Math.Abs(b - a) >= eps2);
l = (a + b) / 2;
//////////

x_new.Clear();
x_new =
Vector<double>.Build.DenseOfVector(x + l * d);
    } while (((x_new - x).L2Norm() > eps3) &&
(iter < 1000));
    return x;
}

}

public class Complex_Method : MethodFields,
IStrategy

```

```

{
    public Complex_Method(Func<Vector<double>,
double> _f, List<Func<Vector<double>, double>> _g,
Vector<double> _x) : base(_f, _g, _x)
    { }

    public Vector<double> Center(Matrix<double>
matr, int num /*количество нужных векторов*/)
    {
        Vector<double> res =
Vector<double>.Build.Dense(matr.RowCount);
        double s = 0;
        for (int i = 0; i < matr.RowCount; i++)
        {
            s = 0;
            for (int j = 0; j < num; j++)
            {
                s += matr[i, j];
            }
            res[i] = s / num;
        }
        return res;
    }

    public Vector<double> Optimize()
    {
        double alpha = 1.07;
        double eps1 = 0.00001; // для f
        double eps2 = 0.00001; // для x
        int p = 5 * n;
        int stop = 0;
        Matrix<double> complex =
Matrix<double>.Build.Dense(n, p);
        for (int i = 0; i < n; i++)
        {
            complex[i, 0] = x[i]; // допустимая
            точка
        }
        Random rand = new Random();
        // Шаг 1
        Vector<double> vec =
Vector<double>.Build.Dense(n);
        int iter = 0;
        do
        {
            stop = 0;
            for (int i = 1; i < p; i++)
            {
                for (int j = 0; j < n; j++)
                {
                    vec[j] = rand.NextDouble();
                }
                if (!MathFunctions.Good_X(g,
vec))
                {
                    int iter1 = 0;
                    do
                    {
                        if ((vec - 0.5 *
(Center(complex, i) + vec)).L2Norm() < 0.00001)
                        {
                            stop++;
                            break;
                        }
                        vec = 0.5 *
(Center(complex, i) + vec);
                        iter1++;
                    } while
(!MathFunctions.Good_X(g, vec) && (iter1 < 100));
                    if (stop > 0) break;
                }
                for (int j = 0; j < n; j++)
                {
                    complex[j, i] = vec[j];
                }
            }
        }
    }
}

```

```

        }
        iter++;
    } while ((stop > 0) && (iter < 1000));
    vec.Clear();
    Vector<double> f_p =
Vector<double>.Build.Dense(p); // f(xp)
    for (int i = 0; i < p; i++)
    {
        f_p[i] = f(complex.Column(i));
    }
    double s1 = 0;
    double s2 = 0;
    iter = 0;
    do
    {
        s1 = 0;
        s2 = 0;
        //War 2
        double Fmax = f_p.Max();
        Vector<double> x_R =
Vector<double>.Build.DenseOfVector(complex.Column(f_p
.MaximumIndex()));
        Vector<double> x_m =
Vector<double>.Build.DenseOfVector((1 + alpha) *
Center(complex, p) - alpha * x_R);
        if (!MathFunctions.Good_X(g, x_m) &&
(f(x_m) >= Fmax))
        {
            int iter1 = 0;
            do
            {
                for (int i = 0; i < n; i++)
                {
                    x_m[i] = (Center(complex,
p)[i] + x_m[i]) / 2;
                }
                iter1++;
            } while ((f(x_m) >= Fmax) &&
(iter1 < 100));
        }
        if (!MathFunctions.Good_X(g, x_m))
        {
            // War 3
            for (int i = 0; i < n; i++)
            {
                if (x_m[i] < 0) x_m[i] = 0;
                if (x_m[i] > 100) x_m[i] =
100;
            }
            int iter1 = 0;
            do
            {
                for (int i = 0; i < n; i++)
                {
                    x_m[i] = (Center(complex,
p)[i] + x_m[i]) / 2;
                }
                iter1++;
            } while (!MathFunctions.Good_X(g,
x_m) && (iter1 < 100));
            for (int i = 0; i < n; i++)
            {
                complex[i, f_p.MaximumIndex()] =
x_m[i];
            }
            for (int i = 0; i < p; i++)
            {
                f_p[i] = f(complex.Column(i));
            }
            // War 4
            for (int i = 0; i < p; i++)
            {

```

```

                s1 += Math.Pow(f_p[i] -
f_p.Average(), 2);
            }
            for (int i = 0; i < p; i++)
            {
                s2 += Math.Pow((complex.Column(i)
- Center(complex, p)).L2Norm(), 2);
            }
            iter++;
        } while (((s1 > eps1) || (s2 > eps2)) &&
(iter < 3000));

        Vector<double> x_new =
Vector<double>.Build.DenseOfVector(Center(complex,
p));
        return x_new;
    }
}

public class Multipliers_Method : MethodFields,
IStrategy
{
    public
Multipliers_Method(Func<Vector<double>, double> _f,
List<Func<Vector<double>, double>> _g,
Vector<double> _x) : base(_f, _g, _x)
    { }

    public double
Min_Restriction(List<Func<Vector<double>, double>> g,
Vector<double> x)
    {
        double s = 0;
        int m = g.Count();
        for (int i = 0; i < m; i++)
        {
            if (s > g[i](x)) s = g[i](x);
        }
        return s;
    }

    public double
Full_Func_for_multipliers_method(Vector<double> x)
    {
        double s = 0;
        int m = g.Count();
        for (int i = 0; i < m; i++)
        {
            s += R[i] *
(Math.Pow(MathFunctions.Cut(g[i](x) + sigma[i]), 2) -
Math.Pow(sigma[i], 2));
        }
        return f(x) + s;
    }

    public Vector<double> Optimize()
    {
        double eps1 = 0.00001; // для одномерной
        double eps2 = 0.001; // для многомерной
        double eps3 = 0.00015; // для штрафной
функции
        double eps4 = 0.001; // для проверки
ограничений
        Vector<double> x_new =
Vector<double>.Build.DenseOfVector(x);
        R = Vector<double>.Build.Dense(m);
        for (int i = 0; i < m; i++)
        {
            R[i] = 10;
        }
        sigma = Vector<double>.Build.Dense(m);
        for (int i = 0; i < m; i++)
        {
            sigma[i] = 10;
        }
    }
}

```

```

double f_old = 0, f_new = 0;
int iter = 0;
do
{
    x.Clear();
    x =
Vector<double>.Build.DenseOfVector(x_new);
    x_new.Clear();
    x_new =
Vector<double>.Build.DenseOfVector(MathFunctions.Conj
ugate_directions_method(
        Full_Func_for_multipliers_method,
x, eps1, eps2, h));
    f_old = f(x);
    if (iter > 0)
    {
        for (int i = 0; i < m; i++)
        {
            sigma[i] =
MathFunctions.Cut(g[i](x) + sigma[i]);
        }
        f_new = f(x_new);
        iter++;
    } while (((Math.Abs(f_new - f_old) >=
eps3) || (Min_Restriction(g, x) < -eps4)) &&
(sigma.L2Norm() > 0.0001) && (iter < 3000));
    return x;
}

public class Penalty_Method : MethodFields,
IStrategy
{
    public Penalty_Method(Func<Vector<double>,
double> _f, List<Func<Vector<double>, double>> _g,
Vector<double> _x) : base(_f, _g, _x)
    { }

    public double
Full_Func_for_penalty_method(Vector<double> x)
    {
        double s = 0;
        int m = g.Count;
        for (int i = 0; i < m; i++)
        {
            s += R[i] *
MathFunctions.Square_Cut(g[i](x));
        }
        return f(x) + s;
    }

    public Vector<double> Optimize()
    {
        double eps1 = 0.000001; // для одномерной
        double eps2 = 0.001; // для многомерной

```

```

double eps3 = 0.001; // для штрафной
функции
    Vector<double> x_new =
Vector<double>.Build.Dense(n);
    R = Vector<double>.Build.Dense(m);
    for (int i = 0; i < m; i++)
    {
        R[i] = 0;
    }
    double P_old = 0, P_new = 0;
    int iter = 0;
    do
    {
        x_new =
Vector<double>.Build.DenseOfVector(
MathFunctions.Conjugate_directions_method(Full_Func_f
or_penalty_method, x, eps1, eps2, h));
        P_old =
Full_Func_for_penalty_method(x);
        if (iter > 0)
        {
            for (int i = 0; i < m; i++)
            {
                R[i] += 1;
            }
            P_new =
Full_Func_for_penalty_method(x_new);
            x.Clear();
            x =
Vector<double>.Build.DenseOfVector(x_new);
            x_new.Clear();
            iter++;
        } while ((Math.Abs(P_new - P_old) >=
eps3) && (iter < 3000));
        return x;
    }

    public class OptContext
    {
        public IStrategy OptMethod { get; set; }

        public OptContext(IStrategy _method)
        {
            OptMethod = _method;
        }

        public Vector<double> Optimization()
        {
            return OptMethod.Optimize();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace Диссертация.Calculation
{
    class Simplex
    {
        //source - симплекс таблица без базисных
переменных
        double[,] table; //симплекс таблица

        int m, n;

```

```

List<int> basis; //список базисных переменных

public Simplex(double[,] source)
{
    m = source.GetLength(0);
    n = source.GetLength(1);
    table = new double[m, n + m - 1];
    basis = new List<int>();

    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j <
table.GetLength(1); j++)
        {
            if (j < n)

```



```

        table[i, j] = source[i, j];
    else
        table[i, j] = 0;
    }
    //выставляем коэффициент 1 перед
    базисной переменной в строке
    if ((n + i) < table.GetLength(1))
    {
        table[i, n + i] = 1;
        basis.Add(n + i);
    }
}

n = table.GetLength(1);

//result - в этот массив будут записаны
полученные значения X
public double[,] Calculate(double[] result)
{
    int mainCol, mainRow; //ведущие столбец и
    строка

    while (!IsItEnd())
    {
        mainCol = findMainCol();
        mainRow = findMainRow(mainCol);
        basis[mainRow] = mainCol;

        double[,] new_table = new double[m,
n];

        for (int j = 0; j < n; j++)
            new_table[mainRow, j] =
table[mainRow, j] / table[mainRow, mainCol];

        for (int i = 0; i < m; i++)
        {
            if (i == mainRow)
                continue;

            for (int j = 0; j < n; j++)
                new_table[i, j] = table[i, j]
- table[i, mainCol] * new_table[mainRow, j];
        }
        table = new_table;
    }

    //записываем в result найденные значения X
    for (int i = 0; i < result.Length; i++)
    {
        int k = basis.IndexOf(i + 1);
        if (k != -1)
            result[i] = table[k, 0];
        else

```

```

        result[i] = 0;
    }
}

return table;

private bool IsItEnd()
{
    bool flag = true;

    for (int j = 1; j < n; j++)
    {
        if (table[m - 1, j] < 0)
        {
            flag = false;
            break;
        }
    }

    return flag;
}

private int findMainCol()
{
    int mainCol = 1;

    for (int j = 2; j < n; j++)
        if (table[m - 1, j] < table[m - 1,
mainCol])
            mainCol = j;

    return mainCol;
}

private int findMainRow(int mainCol)
{
    int mainRow = 0;

    for (int i = 0; i < m - 1; i++)
        if (table[i, mainCol] > 0)
        {
            mainRow = i;
            break;
        }

    for (int i = mainRow + 1; i < m - 1; i++)
        if ((table[i, mainCol] > 0) &&
((table[i, 0] / table[i, mainCol]) < (table[mainRow,
0] / table[mainRow, mainCol])))
            mainRow = i;

    return mainRow;
}
}
}

```

```

.navbar-inverse {
    background-color: #943357;
    border-color: #ffffff;
}

.navbar-inverse .navbar-brand {
    color: #ffffff;
}

.navbar-inverse .navbar-brand:hover,
.navbar-inverse .navbar-brand:focus {
    color: #d2133d;
    background-color: transparent;
}

.navbar-inverse .navbar-text {

```

```

    color: #ffffff;
}

.navbar-inverse .navbar-brand:hover,
.navbar-inverse .navbar-brand:focus {
    color: #d2133d;
    background-color: transparent;
}

.navbar-inverse .navbar-text {
    color: #ffffff;
}

.navbar-inverse .navbar-nav > li > a {
    color: #ffffff;
}

```

```

.navbar-header a.navbar-brand:hover,
.navbar-header a.navbar-brand:focus,
.navbar-inverse .navbar-nav > li > a:hover,
.navbar-inverse .navbar-nav > li > a:focus {
  color: #d2133d;
  background-color: transparent;
  outline: none;
}

.navbar-inverse .navbar-nav > .active > a,
.navbar-inverse .navbar-nav > .active > a:hover,
.navbar-inverse .navbar-nav > .active > a:focus {
  color: #d2133d;
  background-color: #ffffff;
}

.navbar-inverse .navbar-toggle {
  border-color: #ffffff;
}

.navbar-inverse .navbar-toggle:hover,
.navbar-inverse .navbar-toggle:focus {
  color: #d2133d;
  background-color: #ffffff;
  border-color: #d2133d;
}

.navbar-inverse .navbar-toggle:hover .icon-bar,
.navbar-inverse .navbar-toggle:focus .icon-bar {
  color: #d2133d;
  background-color: #d2133d;
  border-color: #d2133d;
}

.navbar-inverse .navbar-nav > .open > a,
.navbar-inverse .navbar-nav > .open > a:hover,
.navbar-inverse .navbar-nav > .open > a:focus {
  color: #d2133d;
  background-color: #943357;
}

.navbar-inverse .navbar-nav > .dropdown > a:hover
.caret {
  border-top-color: #d2133d;
  border-bottom-color: #d2133d;
}

.navbar-inverse .navbar-nav > .dropdown > a .caret {
  border-top-color: #ffffff;
  border-bottom-color: #ffffff;
}

.navbar-inverse .navbar-nav > .open > a .caret,
.navbar-inverse .navbar-nav > .open > a:hover .caret,
.navbar-inverse .navbar-nav > .open > a:focus .caret {
  border-top-color: #d2133d;
  border-bottom-color: #d2133d;
}

.navbar-inverse .navbar-link {
  color: #ffffff;
}

.navbar-inverse .navbar-link:hover {
  color: #d2133d;
}

@media (min-width: 767px) {
  .dropdown:hover .dropdown-menu {
    display: block;
  }
}

```

```

@media (max-width: 767px) {
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > .dropdown-header {
    border-color: #ffffff;
  }
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > li > a {
    color: #ffffff;
  }
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > li > a:hover,
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > li > a:focus {
    color: #d2133d;
    background-color: #943357;
    outline: none;
  }
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > .active > a,
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > .active > a:hover,
  .navbar-inverse .navbar-nav .open .dropdown-menu
  > .active > a:focus {
    color: #d2133d;
    background-color: #ffffff;
    outline: none;
  }
}

.btn:focus {
  outline: none;
}

.btn-primary {
  color: #ffffff;
  background-color: #943357;
  border-color: #000000;
}

.btn-primary:hover,
.btn-primary:focus,
.btn-primary:active,
.btn-primary.active,
.open .dropdown-toggle.btn-primary {
  color: #d2133d;
  background-color: #ffffff;
  border-color: #000000;
}

.btn-primary:active,
.btn-primary.active,
.open .dropdown-toggle.btn-primary {
  background-image: none;
}

.btn-primary.disabled,
.btn-primary[disabled],
fieldset[disabled] .btn-primary,
.btn-primary.disabled:hover,
.btn-primary[disabled]:hover,
fieldset[disabled] .btn-primary:hover,
.btn-primary.disabled:focus,
.btn-primary[disabled]:focus,
fieldset[disabled] .btn-primary:focus,
.btn-primary.disabled:active,
.btn-primary[disabled]:active,
fieldset[disabled] .btn-primary:active,
.btn-primary.disabled.active,
.btn-primary[disabled].active,
fieldset[disabled] .btn-primary.active {
  background-color: #943357;
  border-color: #000000;
}

.btn-link {

```

```

        color: #943357;
    }

    .btn-link:hover,
    .btn-link:focus {
        color: #d2133d;
        text-decoration: underline;
        background-color: #ffffff;
    }

    .form-control:focus {
        border-color: #943357;
        box-shadow: inset 0 1px 1px rgba(159, 70, 124,
0.3);
    }

    .dropdown-menu > li > a {
        color: #000000;
    }

    .dropdown-menu > li > a:hover,
    .dropdown-menu > li > a:focus {

```

```

        color: #ffffff;
        background-color: #943357;
    }
    body {
        padding-top: 50px;
        padding-bottom: 20px;
    }

    .body-content {
        padding-left: 15px;
        padding-right: 15px;
    }

    .dl-horizontal dt {
        white-space: normal;
    }

    input,
    select,
    textarea {
        max-width: 280px;
    }

```

```

using System;
using System.Globalization;
using System.Linq;
using System.Security.Claims;
using System.Threading.Tasks;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using System.Text.RegularExpressions;
using Диссертация.Models;
using Диссертация.Filters;

namespace Диссертация.Controllers
{
    [Culture]
    [Authorize]
    public class AccountController : Controller
    {
        private ApplicationSignInManager
_signInManager;
        private ApplicationUserManager _userManager;

        public AccountController()
        {
        }

        public
AccountController(ApplicationUserManager userManager,
ApplicationSignInManager signInManager )
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }

        public ApplicationSignInManager SignInManager
        {
            get
            {
                return _signInManager ??
HttpContext.GetOwinContext().Get<ApplicationSignInMan
ager>();
            }
            private set
            {
                _signInManager = value;
            }
        }
    }

```

```

        public ApplicationUserManager UserManager
        {
            get
            {
                return _userManager ??
HttpContext.GetOwinContext().GetUserManager<Applicati
onUserManager>();
            }
            private set
            {
                _userManager = value;
            }
        }

        //
        // GET: /Account/Login
        [AllowAnonymous]
        public ActionResult Login(string returnUrl)
        {
            ViewBag.ReturnUrl = returnUrl;
            return View();
        }

        //
        // POST: /Account/Login
        [HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
Login(LoginViewModel model, string returnUrl)
        {
            if (ModelState.IsValid)
            {
                var user = await
UserManager.FindAsync(model.UserName,
model.Password);
                if (user != null)
                {
                    if (user.EmailConfirmed == true)
                    {
                        await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
                        return
RedirectToLocal(returnUrl);
                    }
                    else
                    {

```

```

        ModelState.AddModelError("",
Resources.Resource.UnconfirmedEmail);
    }
    else
    {
        ModelState.AddModelError("",
Resources.Resource.WrongLoginOrPassword);
    }
    return View(model);
}

//
// GET: /Account/VerifyCode
[AllowAnonymous]
public async Task<ActionResult>
VerifyCode(string provider, string returnUrl, bool
rememberMe)
{
    // Require that the user has already
logged in via username/password or external login
    if (!await
SignInManager.HasBeenVerifiedAsync())
    {
        return View("Error");
    }
    return View(new VerifyCodeViewModel {
Provider = provider, ReturnUrl = returnUrl,
RememberMe = rememberMe });
}

//
// POST: /Account/VerifyCode
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
VerifyCode(VerifyCodeViewModel model)
{
    if (!ModelState.IsValid)
    {
        return View(model);
    }

    // The following code protects for brute
force attacks against the two factor codes.
    // If a user enters incorrect codes for a
specified amount of time then the user account
    // will be locked out for a specified
amount of time.
    // You can configure the account lockout
settings in IdentityConfig
    var result = await
SignInManager.TwoFactorSignInAsync(model.Provider,
model.Code, isPersistent: model.RememberMe,
rememberBrowser: model.RememberBrowser);
    switch (result)
    {
        case SignInStatus.Success:
            return
RedirectToLocal(model.ReturnUrl);
        case SignInStatus.LockedOut:
            return View("Lockout");
        case SignInStatus.Failure:
        default:
            ModelState.AddModelError("",
"Invalid code.");
            return View(model);
    }
}

//
// GET: /Account/Register
[AllowAnonymous]

```

```

public ActionResult Register()
{
    return View();
}

//
// POST: /Account/Register
[HttpPost]
[AllowAnonymous]
[ValidateAntiForgeryToken]
public async Task<ActionResult>
Register(RegisterViewModel model)
{
    if (ModelState.IsValid)
    {
        var user = new ApplicationUser {
UserName = model.UserName, Email = model.Email,
DateBirth = TryParseDatetime(model.DateBirth)};
        var result = await
UserManager.CreateAsync(user, model.Password);
        if (result.Succeeded)
        {
            // генерируем токен для
подтверждения регистрации
            var code = await
UserManager.GenerateEmailConfirmationTokenAsync(user.
Id);
            // создаем ссылку для
подтверждения
            var callbackUrl =
Url.Action("ConfirmEmail", "Account", new { userId =
user.Id, code = code },
protocol:
Request.Url.Scheme);
            // отправка письма
            await
UserManager.SendEmailAsync(user.Id,
Resources.Resource.ConfirmationUsingEmail,
Resources.Resource.EmailConfirmationMessageBeg + "
<a href=\""
+ callbackUrl + "\">" +
Resources.Resource.EmailConfirmationMessageEnd +
"</a>");
            return View("DisplayEmail");
        }
        AddErrors(result);
    }
    return View(model);
}

//
// GET: /Account/ConfirmEmail
[AllowAnonymous]
public async Task<ActionResult>
ConfirmEmail(string userId, string code)
{
    if (userId == null || code == null)
    {
        return View("Error");
    }
    var result = await
UserManager.ConfirmEmailAsync(userId, code);

    ApplicationDbContext db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>
();
    foreach (var met in db.Methods.Select(m
=> new { MethodType = m.MethodType })).Distinct())
    //Where(m => m.Language == 1))
    {
        db.MethodSettings.Add(new
MethodSetting { Usage = false, MethodType =
met.MethodType, UserId = userId });
    }
}

```

```

        }
        foreach (var func in
db.Functions.Select(f => new { FunctionType =
f.FunctionType }).Distinct()) //Where(l => l.Language
== 1))
        {
            db.FunctionSettings.Add(new
FunctionSetting { Limitation = false, Criteria =
false, FunctionType = func.FunctionType, UserId =
userId });
        }
        db.SaveChanges();
        return View(result.Succeeded ?
"ConfirmEmail" : "Error");
    }

    //
    // GET: /Account/ForgotPassword
    [AllowAnonymous]
    public ActionResult ForgotPassword()
    {
        return View();
    }

    //
    // POST: /Account/ForgotPassword
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
ForgotPassword(ForgotPasswordViewModel model)
    {
        if (ModelState.IsValid)
        {
            var user = await
userManager.FindByEmailAsync(model.Email);
            if (user == null || !(await
userManager.IsEmailConfirmedAsync(user.Id)))
            {
                return
View("ForgotPasswordConfirmation");
            }
            string code = await
userManager.GeneratePasswordResetTokenAsync(user.Id);
            var callbackUrl =
Url.Action("ResetPassword", "Account",
            new { userId = user.Id, code =
code }, protocol: Request.Url.Scheme);
            await
userManager.SendEmailAsync(user.Id,
Resources.Resource.ResetPassword,

Resources.Resource.EmailResetPasswordMessageBeg +
user.UserName +
Resources.Resource.EmailResetPasswordMessageMid +
": <a href=\"" + callbackUrl +
"\>" +
Resources.Resource.EmailResetPasswordMessageEnd
+ "</a>");
            return
View("ForgotPasswordConfirmation");
        }
        return View(model);
    }

    //
    // GET: /Account/ForgotPasswordConfirmation
    [AllowAnonymous]
    public ActionResult
ForgotPasswordConfirmation()
    {
        return View();
    }

    //

```

```

    // GET: /Account/ResetPassword
    [AllowAnonymous]
    public ActionResult ResetPassword(string
code)
    {
        return code == null ? View("Error") :
View();
    }

    //
    // POST: /Account/ResetPassword
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
ResetPassword(ResetPasswordViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        var user = await
userManager.FindByEmailAsync(model.Email);
        if (user == null)
        {
            // Don't reveal that the user does
not exist
            return
RedirectToAction("ResetPasswordConfirmation",
"Account");
        }
        var result = await
userManager.ResetPasswordAsync(user.Id, model.Code,
model.Password);
        if (result.Succeeded)
        {
            return
RedirectToAction("ResetPasswordConfirmation",
"Account");
        }
        AddErrors(result);
        return View();
    }

    //
    // GET: /Account/ResetPasswordConfirmation
    [AllowAnonymous]
    public ActionResult
ResetPasswordConfirmation()
    {
        return View();
    }

    //
    // POST: /Account/ExternalLogin
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public ActionResult ExternalLogin(string
provider, string returnUrl)
    {
        // Request a redirect to the external
login provider
        return new ChallengeResult(provider,
Url.Action("ExternalLoginCallback", "Account", new {
ReturnUrl = returnUrl }));
    }

    //
    // GET: /Account/SendCode
    [AllowAnonymous]
    public async Task<ActionResult>
SendCode(string returnUrl, bool rememberMe)
    {

```

```

        var userId = await
SignInManager.GetVerifiedUserIdAsync();
        if (userId == null)
        {
            return View("Error");
        }
        var userFactors = await
userManager.GetValidTwoFactorProvidersAsync(userId);
        var factorOptions =
userFactors.Select(purpose => new SelectListItem {
Text = purpose, Value = purpose }).ToList();
        return View(new SendCodeViewModel {
Providers = factorOptions, returnUrl = returnUrl,
RememberMe = rememberMe });
    }

    //
    // POST: /Account/SendCode
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
SendCode(SendCodeViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View();
        }

        // Generate the token and send it
        if (!await
SignInManager.SendTwoFactorCodeAsync(model.SelectedPr
ovider))
        {
            return View("Error");
        }
        return RedirectToAction("VerifyCode", new
{ Provider = model.SelectedProvider, returnUrl =
model.ReturnUrl, RememberMe = model.RememberMe });
    }

    //
    // GET: /Account/ExternalLoginCallback
    [AllowAnonymous]
    public async Task<ActionResult>
ExternalLoginCallback(string returnUrl)
    {
        var loginInfo = await
AuthenticationManager.GetExternalLoginInfoAsync();
        if (loginInfo == null)
        {
            return RedirectToAction("Login");
        }

        // Sign in the user with this external
login provider if the user already has a login
        var result = await
SignInManager.ExternalSignInAsync(loginInfo,
isPersistent: false);
        switch (result)
        {
            case SignInStatus.Success:
                return
RedirectToLocal(returnUrl);
            case SignInStatus.LockedOut:
                return View("Lockout");
            case
SignInStatus.RequiresVerification:
                return
RedirectToAction("SendCode", new { returnUrl =
returnUrl, RememberMe = false });
            case SignInStatus.Failure:
            default:
                // If the user does not have an
account, then prompt the user to create an account

```

```

        ViewBag.ReturnUrl = returnUrl;
        ViewBag.LoginProvider =
loginInfo.Login.LoginProvider;
        return
View("ExternalLoginConfirmation", new
ExternalLoginConfirmationViewModel { Email =
loginInfo.Email });
    }

    //
    // POST: /Account/ExternalLoginConfirmation
    [HttpPost]
    [AllowAnonymous]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
ExternalLoginConfirmation(ExternalLoginConfirmationVi
ewModel model, string returnUrl)
    {
        if (User.Identity.IsAuthenticated)
        {
            return RedirectToAction("MainMenu",
"Manage");
        }

        if (ModelState.IsValid)
        {
            // Get the information about the user
from the external login provider
            var info = await
AuthenticationManager.GetExternalLoginInfoAsync();
            if (info == null)
            {
                return
View("ExternalLoginFailure");
            }
            var user = new ApplicationUser {
UserName = model.Email, Email = model.Email };
            var result = await
userManager.CreateAsync(user);
            if (result.Succeeded)
            {
                result = await
userManager.AddLoginAsync(user.Id, info.Login);
                if (result.Succeeded)
                {
                    await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
                    return
RedirectToLocal(returnUrl);
                }
                AddErrors(result);
            }

            ViewBag.ReturnUrl = returnUrl;
            return View(model);
        }

        //
        // POST: /Account/LogOff
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult LogOff()
        {
            AuthenticationManager.SignOut(DefaultAuthenticationTy
pes.ApplicationCookie);
            return RedirectToAction("MainMenu",
"Home");
        }

        //
        // GET: /Account/ExternalLoginFailure

```

```

[AllowAnonymous]
public ActionResult ExternalLoginFailure()
{
    return View();
}

protected override void Dispose(bool
disposing)
{
    if (disposing)
    {
        if (_userManager != null)
        {
            _userManager.Dispose();
            _userManager = null;
        }

        if (_signInManager != null)
        {
            _signInManager.Dispose();
            _signInManager = null;
        }
    }

    base.Dispose(disposing);
}

#region Helpers
// Used for XSRF protection when adding
external logins
private const string XsrfKey = "XsrfId";

private IAuthenticationManager
AuthenticationManager
{
    get
    {
        return
HttpContext.GetOwinContext().Authentication;
    }
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        if (error.StartsWith("Name"))
        {
            var new_error =
Resources.Resource.ErrorDuplicateLogin;
ModelState.AddModelError("",
new_error);
        }
        else if (error.StartsWith("Email"))
        {
            var new_error =
Resources.Resource.ErrorDuplicateEmail;
ModelState.AddModelError("",
new_error);
        }
        else if (error.StartsWith("Passwords
must have at least one"))
        {
            var new_error =
Resources.Resource.ErrorPasswordFormat;
ModelState.AddModelError("",
new_error);
        }
        else
    }
}

    {
        ModelState.AddModelError("",
error);
    }
}

private ActionResult RedirectToLocal(string
returnUrl)
{
    if (Url.IsLocalUrl(returnUrl))
    {
        return Redirect(returnUrl);
    }
    return RedirectToAction("MainMenu",
"Home");
}

private DateTime? TryParseDatetime(string
value)
{
    DateTime db;
    DateTime.TryParse(value, out db);
    if (db >= new DateTime(1900, 1, 1))
        return db;
    else
        return null;
}

internal class ChallengeResult :
HttpUnauthorizedResult
{
    public ChallengeResult(string provider,
string redirectUri)
        : this(provider, redirectUri, null)
    {
    }

    public ChallengeResult(string provider,
string redirectUri, string userId)
    {
        LoginProvider = provider;
        RedirectUri = redirectUri;
        UserId = userId;
    }

    public string LoginProvider { get; set; }
    public string RedirectUri { get; set; }
    public string UserId { get; set; }

    public override void
ExecuteResult(ControllerContext context)
    {
        var properties = new
AuthenticationProperties { RedirectUri = RedirectUri
};
        if (UserId != null)
        {
            properties.Dictionary[XsrfKey] =
UserId;
        }

        context.HttpContext.GetOwinContext().Authentication.C
hallenge(properties, LoginProvider);
    }
}
#endregion
}
}

```

```

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Web;

```

```
using System.Web.Mvc;
using Диссертация.Filters;
```

```
namespace Диссертация.Controllers
{
    [Culture]
    public class ErrorController : Controller
    {
        public ActionResult Unauthorized()
        {
            Response.StatusCode = 401;
            return View();
        }
    }
}
```

```
public ActionResult NotFound()
{
    Response.StatusCode = 404;
    return View();
}

public ActionResult InternalServerError()
{
    Response.StatusCode = 500;
    return View();
}
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Globalization;
using System.Web;
using System.Web.Mvc;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using MathNet.Numerics;
using MathNet.Numerics.LinearAlgebra;
using Диссертация.Filters;
using Диссертация.Models;
using Диссертация.Calculation;
using Microsoft.AspNet;
using Newtonsoft.Json;

namespace Диссертация.Controllers
{
    [Culture]
    public class HomeController : Controller
    {
        const int million = 1000000;
        const int NotFoundDot = 0; // точка не
найдена
        const int FoundNewDot = 1; // точка найдена с
помощью перебора
        const int FoundArchievedDot = 2; // точка
найдена из базы
        const int n = 7; // количество переменных
геометрических частей двигателя
        const int m = 8; // количество всех функций
характеристик двигателя
        const int Sign_Pos = 1, Sign_Neg = 2; //1-
положительный знак, 2-отрицательный знак

        private ApplicationDbContext db;
        private Func<Vector<double>, double> f;
        private int f_Sign;
        private List<Func<Vector<double>, double>> g;
        private Vector<double> x;
        private Vector<double> b_vec;
        private static List<ResultsViewModel>
resengines = new List<ResultsViewModel>();

        public HomeController()
        {
            g = new List<Func<Vector<double>,
double>>();
            x = Vector<double>.Build.Dense(n, 0);
            b_vec = Vector<double>.Build.Dense(m, 0);
        }

        [Authorize]
        public ActionResult MainMenu(List<double>
lst, int? dot_find_met, int? best_met)
```

```

        {
            db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>
>();
            if (db.MethodSettings.Where(m =>
m.User.UserName == User.Identity.Name).Where(m =>
m.Usage == true).Count() == 0)
                return RedirectToAction("Settings",
new { Message = SettingsMessageId.NeedSetMethods });
            if (db.FunctionSettings.Where(f =>
f.User.UserName == User.Identity.Name).Where(f =>
f.Limitation == true).Count() == 0)
                return RedirectToAction("Settings",
new { Message = SettingsMessageId.NeedSetLimitations
});
            if (db.FunctionSettings.Where(f =>
f.User.UserName == User.Identity.Name).Where(f =>
f.Criteria == true).Count() == 0)
                return RedirectToAction("Settings",
new { Message = SettingsMessageId.NeedSetCriteria });

            if (lst != null)
            {
                Vector<double> vec =
MathFunctions.list_to_vector(lst);
                for (int i = 0; i < n; i++)
                {
                    x[i] = vec[i];
                }
                for(int i = n; i < vec.Count;i++)
                {
                    b_vec[i - n] = vec[i];
                }
            }
            var model = new MainMenuViewModel
            {
                Limitation = GetMenuLimitations(),
                Argument = GetMenuArguments(),
                ResultText = ""
            };
            if (lst != null )
            {
                if (!MathFunctions.IsZeroVector(x))
                {
                    model.ResultText =
Resources.Resource.ResultEngineHeader + "\n";
                    var criterias = from
functionsettings in db.FunctionSettings
                                join cri in
db.Functions on functionsettings.FunctionType equals
cri.FunctionType
                                join languages in
db.Languages on cri.LanguageId equals languages.Id
                                where
languages.LCID == CultureInfo.CurrentCulture.LCID
```



```

        where
functionsettings.User.UserName == User.Identity.Name
        where
functionsettings.Criteria == true
        select new
        {
            cri.FunctionType,
            cri.NameLong
        };
        model.ResultText +=
        criterias.First().NameLong + " = " +
        Functions.FindFunction(criterias.First().FunctionType
        )(x).ToString().Replace(".", ",") + "\n";
        var limitations = from
        functionsettings in db.FunctionSettings
        join lim in
        db.Functions on functionsettings.FunctionType equals
        lim.FunctionType
        join languages
        in db.Languages on lim.LanguageId equals languages.Id
        where
        languages.LCID == CultureInfo.CurrentCulture.LCID
        where
        functionsettings.User.UserName == User.Identity.Name
        where
        functionsettings.Limitation == true
        orderby
        lim.FunctionType
        select new
        {
            lim.FunctionType,
            lim.NameLong
        };
        foreach (var lim in limitations)
        {
            model.ResultText +=
            lim.NameLong + " = " +
            Functions.FindFunction(lim.FunctionType)(x).ToString(
            ).Replace(".", ",") + "\n";
        }
        if (dot_find_met ==
        FoundArchievedDot)
            model.ResultText +=
            Resources.Resource.FoundArchievedDot + "\n";
        else if (dot_find_met ==
        FoundNewDot)
            model.ResultText +=
            Resources.Resource.FoundNewDot + "\n";
            if (best_met == 0)
                model.ResultText +=
                Resources.Resource.NoAnswer;
            else
                model.ResultText +=
                Resources.Resource.BestMethodHeader + " " +
                FuncBestMethod((int)best_met);
        }
        else
        {
            model.ResultText =
            Resources.Resource.EmptySet;
        }
    }

    return View(model);
}

private string FuncBestMethod( int best_met )
{
    var language = from languages in
db.Languages
        where languages.LCID ==
CultureInfo.CurrentCulture.LCID

```

```

        select new
        {
            languages.Id
        };
        int l = 0;
        foreach (var lang in language)
        {
            l = lang.Id;
        }
        switch( l )
        {
            case 1:
                switch( best_met )
                {
                    case 1:
                        return "метода допустимых
направлений.";
                    case 2:
                        return "метода
комплексов.";
                    case 3:
                        return "метода множителей
Лагранжа.";
                    case 4:
                        return "метода штрафных
функций.";
                }
                break;
            case 2:
                switch( best_met )
                {
                    case 1:
                        return "методу допустимих
напрямків.";
                    case 2:
                        return "методу
комплексів.";
                    case 3:
                        return "методу множників
Лагранжа.";
                    case 4:
                        return "методу штрафних
функцій.";
                }
                break;
            case 3:
                switch( best_met )
                {
                    case 1:
                        return "method of
allowable directions.";
                    case 2:
                        return "method of
complexes.";
                    case 3:
                        return "method of
Lagrange multipliers.";
                    case 4:
                        return "method of penalty
funtions.";
                }
                break;
        }
        return "";
    }

    private IList<MenuItem>
    GetMenuLimitations()
    {
        db =
HttpContext.GetOwinContext().Get<ApplicationDbContext
>();
        var limitations = from functionsettings
in db.FunctionSettings

```

```

        join lim in
db.Functions on functionsettings.FunctionType equals
lim.FunctionType
        join funcval in
db.FunctionValues on functionsettings.FunctionType
equals funcval.Id
        join languages in
db.Languages on lim.LanguageId equals languages.Id
        where languages.LCID ==
CultureInfo.CurrentCulture.LCID
        where
functionsettings.User.UserName == User.Identity.Name
        where
functionsettings.Limitation == true
        orderby
lim.FunctionType
        select new
{
    Id =
functionsettings.FunctionType,
    Usage =
functionsettings.Limitation,
    Name =
lim.NameShort,
    NameDef =
lim.NameLong,
    Sign =
funcval.Sign,
    DefaultValue =
funcval.DefaultValue,
    MinValue =
funcval.MinValue,
    MaxValue =
funcval.MaxValue,
    User =
functionsettings.User.UserName
};
List<MenuItem> listLimitations = new
List<MenuItem>();
int i = 0;
foreach (var lim in limitations)
{
    i++;
    MenuItem l = new MenuItem()
    {
        Id = lim.Id,
        Text = lim.Name,
        Definition = ( i != n ) ?
lim.NameDef + ", " : lim.NameDef,
        Sign = lim.Sign,
        Value =
MathFunctions.IsZeroVector(b_vec) ?

lim.DefaultValue.ToString().Replace(".", ",") :
b_vec[lim.Id-1].ToString().Replace(".", ","),
        MinValue = lim.MinValue,
        MaxValue = lim.MaxValue
    };
    listLimitations.Add(l);
}

return listLimitations;
}

private IList<MenuItem>
GetMenuArguments()
{
    db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>();
    var arguments = from args in db.Arguments
        join languages in
db.Languages on args.LanguageId equals languages.Id
        where languages.LCID ==
CultureInfo.CurrentCulture.LCID

```

```

        orderby args.NameShort
        select new
{
    Id =
args.AgrumentType,
    Name =
args.NameShort,
    NameDef =
args.NameLong
};
bool is_empty_X =
(MathFunctions.IsEmptyVector(x) ||
MathFunctions.IsZeroVector(x));
var listArguments = new
List<MenuItem>();
int i = 0;
foreach (var arg in arguments)
{
    i++;
    MenuItem a = new MenuItem()
    {
        Id = arg.Id,
        Text = arg.Name,
        Definition = ( i != n ) ?
arg.NameDef + ", " : arg.NameDef
    };
    if (!is_empty_X)
    {
        a.Value = x[a.Id -
1].ToString().Replace(".", ",");
        listArguments.Add(a);
    }
    return listArguments;
}

[HttpPost]
public ActionResult
MainMenu(MainMenuViewModel model)
{
    double min_func;
    int best_met = 0, dot_find_met = 0;
    Vector<double> x0 =
Vector<double>.Build.Dense(n, 0);
    Vector<double> x_ans =
Vector<double>.Build.Dense(n, 0);

    db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>();
    for (int i = 0; i <
model.LimitationId.Count; i++)
    {
        g.Add(FindFunction_g(model.LimitationId[i]));
        b_vec[model.LimitationId[i] - 1] =
model.LimitationValue[i];
    }

    var criterias = from functionsettings in
db.FunctionSettings
        join functionvalues in
db.FunctionValues on functionsettings.FunctionType
equals functionvalues.Id
        where
functionsettings.User.UserName == User.Identity.Name
        where
functionsettings.Criteria == true
        select new
{
    functionsettings.FunctionType,
    functionvalues.Sign
};

```

```

        f =
FindFunction_f(criteria.First().FunctionType);
        f_Sign = criteria.First().Sign;

        dot_find_met =
FindStartingPoint(model.LimitationId);
        if (dot_find_met != NotFoundDot )
        {
            var methods = from methodsetting in
db.MethodSettings
                        where
methodsetting.User.UserName == User.Identity.Name
                        where
methodsetting.Usage == true
                        orderby
methodsetting.MethodType
                        select
methodsetting.MethodType;

            min_func = f(x);
            x0 =
Vector<double>.Build.DenseOfVector(x);
            x_ans =
Vector<double>.Build.DenseOfVector(x);
            Task[] tasks = new
Task[methods.Count()];
            int met_iter = 0;
            foreach (int met in methods)
            {
                tasks[met_iter] = new Task(() =>
                {
                    x =
Vector<double>.Build.DenseOfVector(x0);
                    OptContext con = new
OptContext(FindMethod(met));
                    x =
Vector<double>.Build.DenseOfVector(con.Optimization()
);

                    if (MathFunctions.Good_X(g,
x) && MathFunctions.Positive_X(x) &&
MathFunctions.Positive_g(model.LimitationId, g,
b_vec, x) &&
                    ((f_Sign == Sign_Pos) &&
(f(x) > 0)) || ((f_Sign == Sign_Neg) && (f(x) < 0))))
                    {
                        if (min_func > f(x))
                        {
                            min_func = f(x);
                            best_met = met;
                            x_ans =
Vector<double>.Build.DenseOfVector(x);
                        }
                    }
                });
                met_iter++;
            }
            foreach (var t in tasks)
            {
                t.Start();
            }
            Task.WaitAll(tasks);
            x =
Vector<double>.Build.DenseOfVector(x_ans);
            Engine engine = new Engine
            {
                EngineDate = DateTime.Now,
                Method = best_met,
                UserId =
User.Identity.GetUserId()
            };
            db.Engines.Add(engine);
            for (int i = 0; i < n; i++)
            {
                EngineParameter ep = new
EngineParameter

```

```

            {
                EngineId = engine.Id,
                Argument = (i + 1),
                Value = x[i]
            };
            db.EngineParameters.Add(ep);
        }
        EngineCharacteristic ec1 = new
EngineCharacteristic
        {
            EngineId = engine.Id,
            Function =
criteria.First().FunctionType,
            Criteria = true,
            Value =
Functions.FindFunction(criteria.First().FunctionType
)(x)
        };
        db.EngineCharacteristics.Add(ec1);
        for (int i = 0; i <
model.LimitationId.Count; i++)
        {
            EngineCharacteristic ec2 = new
EngineCharacteristic
            {
                EngineId = engine.Id,
                Function =
model.LimitationId[i],
                Criteria = false,
                Value =
Functions.FindFunction(model.LimitationId[i])(x)
            };
            db.EngineCharacteristics.Add(ec2);
        }
        db.SaveChanges();
    }
    else
    {
        x = Vector<double>.Build.Dense(n, 0);
    }
    string str = "";
    if ( x != null )
    {
        str = "lst[0]=" +
x[0].ToString("F10").Replace(",", ".");
        for (int i = 1; i < n; i++ )
        {
            str += "&lst[" + i.ToString() +
"]=" + x[i].ToString("F10").Replace(",", ".");
        }
        for(int i = n; i < (n+m); i++)
        {
            str += "&lst[" + i.ToString() +
"]=" + b_vec[i-n].ToString("F10").Replace(",", ".");
        }
        str += "&dot_find_met=" +
dot_find_met.ToString();
        str += "&best_met=" +
best_met.ToString();
    }
    return
Redirect("/EngineOpt/Home/MainMenu?" + str);
}

private int FindStartingPoint(IList<int>
g_num)
{
    int iter = 0;
    do
    {
        iter++;
        Random rand = new Random();
        x[0] = rand.NextDouble() * 0.05 +
0.05;
    }
    while (true);
}

```

```

        x[1] = rand.NextDouble() * 0.05 +
0.03;
        x[2] = rand.NextDouble() * 0.002 +
0.003;
        x[3] = rand.NextDouble() * 0.002 +
0.0055;
        x[4] = rand.NextDouble() * 0.05 +
0.05;
        x[5] = rand.NextDouble() * 0.01 +
0.014;
        x[6] = rand.NextDouble() * 0.1 +
0.07;
    } while ((!MathFunctions.Good_X(g, x) ||
!MathFunctions.Positive_g(g_num, g, b_vec, x)
    || !(((f_Sign == Sign_Pos) && (f(x) >
0)) || ((f_Sign == Sign_Neg) && (f(x) < 0)))) &&
(iter < million));
    if (iter == million)
    {
        return FindArchivedEngine(g_num);
    }
    else
    {
        return FoundNewDot;
    }
}

private int FindArchivedEngine(IList<int>
g_num)
{
    db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>
>();
    IQueryable<int> engines =
db.Engines.Select(eng => eng.Id );
    for(int i = 0; i < m; i++)
    {
        if (g_num.Contains(i+1))
            engines = FilterEngines(i+1,
engines);
    }
    if (engines.Count() == 0)
        return NotFoundDot;

    int e_id = engines.FirstOrDefault();
    var tmp_x = db.EngineParameters.Where(ep
=> (ep.EngineId == e_id))
        .Select(ep => new { Argument =
ep.Argument, Value = ep.Value });
    for (int i = 0; i < n; i++)
    {
        x[i] = tmp_x.Where(ep => (ep.Argument
== (i + 1))).Select(ep => ep.Value).Max();
    }
    return FoundArchievedDot;
}

private IQueryable<int> FilterEngines(int
func, IQueryable<int> engines)
{
    double restr = b_vec[func - 1];
    db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>
>();
    engines = from eng in engines
                join engch in
db.EngineCharacteristics on new { EngineId = eng,
Function = func }
                equals new { engch.EngineId,
engch.Function }
                join f in db.FunctionValues on
func equals f.Id
                where (((engch.Value >= restr)
&& (f.Sign==2)) || ((engch.Value <= restr) &&
(f.Sign==1)))
                select eng;

    select eng;
    return engines;
}

private IStrategy FindMethod(int n)
{
    switch (n)
    {
        case 1:
            return new
Allowable_directions_Method(f, g, x);
        case 2:
            return new Complex_Method(f, g,
x);
        case 3:
            return new Multipliers_Method(f,
g, x);
        case 4:
            return new Penalty_Method(f,g,x);
        default:
            return new Penalty_Method(f, g,
x);
    }
}

private Func<Vector<double>, double>
FindFunction_f(int n) // Функции для критерия
{
    switch (n)
    {
        case 1:
            return Mmax_f;
        case 2:
            return Mt_f;
        case 3:
            return Mc_f;
        case 4:
            return Gff_f;
        case 5:
            return Cosfi_f;
        case 6:
            return Eff_f;
        case 7:
            return M_f;
        case 8:
            return Tw_f;
        default:
            return Mmax_f;
    }
}

#region Criterias
private double Mmax_f(Vector<double> x)
{
    return -Functions.Mmax(x);
}

private double Mt_f(Vector<double> x)
{
    return -Functions.Mt(x);
}

private double Mc_f(Vector<double> x)
{
    return Functions.Mc(x);
}

private double Gff_f(Vector<double> x)
{
    return -Functions.Gff(x);
}

private double Cosfi_f(Vector<double> x)
{
    return -Functions.Cosfi(x);
}

```

```

    }

    private double Eff_f(Vector<double> x)
    {
        return -Functions.Eff(x);
    }

    private double M_f(Vector<double> x)
    {
        return Functions.M(x);
    }

    private double Tw_f(Vector<double> x)
    {
        return Functions.Tw(x);
    }
    #endregion

    private Func<Vector<double>, double>
    FindFunction_g(int n) // Функции для ограничений
    {
        switch (n)
        {
            case 1:
                return Mmax_g;
            case 2:
                return Mt_g;
            case 3:
                return Mc_g;
            case 4:
                return Gff_g;
            case 5:
                return Cosfi_g;
            case 6:
                return Eff_g;
            case 7:
                return M_g;
            case 8:
                return Tw_g;
            default:
                return Mmax_g;
        }
    }

    #region Restrictions
    private double Mmax_g(Vector<double> x)
    {
        return Functions.Mmax(x) - b_vec[0];
    }

    private double Mt_g(Vector<double> x)
    {
        return Functions.Mt(x) - b_vec[1];
    }

    private double Mc_g(Vector<double> x)
    {
        return b_vec[2] - Functions.Mc(x);
    }

    private double Gff_g(Vector<double> x)
    {
        return Functions.Gff(x) - b_vec[3];
    }

    private double Cosfi_g(Vector<double> x)
    {
        return Functions.Cosfi(x) - b_vec[4];
    }

    private double Eff_g(Vector<double> x)
    {
        return Functions.Eff(x) - b_vec[5];
    }

```

```

        private double M_g(Vector<double> x)
        {
            return b_vec[6] - Functions.M(x);
        }

        private double Tw_g(Vector<double> x)
        {
            return b_vec[7] - Functions.Tw(x);
        }
    #endregion

    [Authorize]
    public ActionResult
    Settings(SettingsMessageId? message)
    {
        ViewBag.StatusMessage =
            message ==
            SettingsMessageId.SaveSettingsSuccess ?
            Resources.Resource.SaveSettingsSuccess
                : message ==
            SettingsMessageId.SaveSettingsError ?
            Resources.Resource.Error
                : message ==
            SettingsMessageId.NeedSetMethods ?
            Resources.Resource.NeedSetMethods
                : message ==
            SettingsMessageId.NeedSetLimitations ?
            Resources.Resource.NeedSetLimitations
                : message ==
            SettingsMessageId.NeedSetCriteria ?
            Resources.Resource.NeedSetCriteria
                : "";

        var model = new SettingsViewModel
        {
            MethodsList = GetMethods(),
            LimitationsList = GetLimitations(),
            CriteriasList = GetCriterias()
        };
        return View(model);
    }

    [HttpPost]
    public async Task<ActionResult>
    Settings(SettingsViewModel model)
    {
        db =
            HttpContext.GetOwinContext().Get<ApplicationDbContext>();

        foreach (var met in
            db.MethodSettings.Where(m => m.User.UserName ==
            User.Identity.Name))
        {
            met.Usage = false;
        }
        foreach (int i in model.SelectedMethods)
        {
            MethodSetting met = await
            db.MethodSettings.FindAsync(i);
            met.Usage = true;
        }
        foreach (var lim in
            db.FunctionSettings.Where(m => m.User.UserName ==
            User.Identity.Name))
        {
            lim.Limitation = false;
        }
        foreach (int i in
            model.SelectedLimitations)
        {
            FunctionSetting lim = await
            db.FunctionSettings.FindAsync(i);
            lim.Limitation = true;
        }
    }

```

```

        foreach (var cri in
db.FunctionSettings.Where(m => m.User.UserName ==
User.Identity.Name))
        {
            cri.Criteria = false;
        }
        foreach (int i in
model.SelectedCriteria)
        {
            FunctionSetting cri = await
db.FunctionSettings.FindAsync(i);
            cri.Criteria = true;
        }
        int ret = db.SaveChanges();
        if (ret != 0)
        {
            return RedirectToAction("Settings",
new { Message = SettingsMessageId.SaveSettingsSuccess
});
        }
        else
        {
            return RedirectToAction("Settings",
new { Message = SettingsMessageId.SaveSettingsError
});
        }
    }

    private IList<SelectListItem> GetMethods()
    {
        db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>();
        var methods = from methodsettings in
db.MethodSettings
                        join met in db.Methods on
methodsettings.MethodType equals met.MethodType
                        join languages in
db.Languages on met.LanguageId equals languages.Id
                        where languages.LCID ==
CultureInfo.CurrentCulture.LCID
                        where
methodsettings.User.UserName == User.Identity.Name
                        orderby met.Name
                        select new
                        {
                            Id = methodsettings.Id,
                            Usage =
methodsettings.Usage,
                            Name = met.Name,
                            User =
methodsettings.User.UserName
                        };
        var checkBoxListMethods = new
List<SelectListItem>();
        foreach (var met in methods)
        {
            SelectListItem cb = new
SelectListItem()
            {
                Value = met.Id.ToString(),
                Text = met.Name,
                Selected = met.Usage
            };
            checkBoxListMethods.Add(cb);
        }
        return checkBoxListMethods;
    }

    private IList<SelectListItem>
GetLimitations()
    {
        db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>();

```

```

        var limitations = from functionsettings
in db.FunctionSettings
                        join lim in
db.Functions on functionsettings.FunctionType equals
lim.FunctionType
                        join languages in
db.Languages on lim.LanguageId equals languages.Id
                        where languages.LCID ==
CultureInfo.CurrentCulture.LCID
                        where
functionsettings.User.UserName == User.Identity.Name
                        orderby lim.NameLong
                        select new
                        {
                            Id =
functionsettings.Id,
                            Usage =
functionsettings.Limitation,
                            Name =
lim.NameLong,
                            User =
functionsettings.User.UserName
                        };
        var checkBoxListLimitations = new
List<SelectListItem>();
        foreach (var lim in limitations)
        {
            SelectListItem cb = new
SelectListItem()
            {
                Value = lim.Id.ToString(),
                Text = lim.Name,
                Selected = lim.Usage
            };
            checkBoxListLimitations.Add(cb);
        }
        return checkBoxListLimitations;
    }

    private IList<SelectListItem> GetCriteria()
    {
        db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>();
        var criterias = from functionsettings in
db.FunctionSettings
                        join crit in db.Functions
on functionsettings.FunctionType equals
crit.FunctionType
                        join languages in
db.Languages on crit.LanguageId equals languages.Id
                        where languages.LCID ==
CultureInfo.CurrentCulture.LCID
                        where
functionsettings.User.UserName == User.Identity.Name
                        orderby crit.NameLong
                        select new
                        {
                            Id =
functionsettings.Id,
                            Usage =
functionsettings.Criteria,
                            Name = crit.NameLong,
                            User =
functionsettings.User.UserName
                        };
        var checkBoxListCriteria = new
List<SelectListItem>();
        foreach (var cri in criterias)
        {
            SelectListItem cb = new
SelectListItem()
            {
                Value = cri.Id.ToString(),
                Text = cri.Name,

```

```

        Selected = cri.Usage
    };
    checkBoxListCriteriaas.Add(cb);
}
return checkBoxListCriteriaas;
}

[Authorize]
public ActionResult Results()
{
    db =
HttpContext.GetOwinContext().Get<ApplicationDbContext>
>();
    List<ResultsArgsModel> resa = new
List<ResultsArgsModel>();
    List<ResultsFuncModel> resf = new
List<ResultsFuncModel>();
    resengines.Clear();
    var resargs = from engines in db.Engines
join arg1 in
db.EngineParameters on engines.Id equals
arg1.EngineId
join arg2 in
db.EngineParameters on engines.Id equals
arg2.EngineId
join arg3 in
db.EngineParameters on engines.Id equals
arg3.EngineId
join arg4 in
db.EngineParameters on engines.Id equals
arg4.EngineId
join arg5 in
db.EngineParameters on engines.Id equals
arg5.EngineId
join arg6 in
db.EngineParameters on engines.Id equals
arg6.EngineId
join arg7 in
db.EngineParameters on engines.Id equals
arg7.EngineId
where engines.User.UserName
== User.Identity.Name
where arg1.Argument == 1
where arg2.Argument == 2
where arg3.Argument == 3
where arg4.Argument == 4
where arg5.Argument == 5
where arg6.Argument == 6
where arg7.Argument == 7
select new
{
    Id = engines.Id,
    Date =
engines.EngineDate,
    Arg_Ds = arg1.Value,
    Arg_Ls = arg2.Value,
    Arg_bzs = arg3.Value,
    Arg_bzr = arg4.Value,
    Arg_has = arg5.Value,
    Arg_har = arg6.Value,
    Arg_Da = arg7.Value
};
foreach(var r in resargs)
{
    resa.Add(new ResultsArgsModel
    {
        Id = r.Id,
        Date = r.Date,
        Arg_Ds = r.Arg_Ds,
        Arg_Ls = r.Arg_Ls,
        Arg_bzs = r.Arg_bzs,
        Arg_bzr = r.Arg_bzr,
        Arg_has = r.Arg_has,
        Arg_har = r.Arg_har,
        Arg_Da = r.Arg_Da
    });
}
}

```

```

    });
}

var resfunc = from engines in db.Engines
join fun1 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 1 }
equals new {
fun1.EngineId, fun1.Function } into f1
from t1 in
f1.DefaultIfEmpty()
join fun2 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 2 }
equals new {
fun2.EngineId, fun2.Function } into f2
from t2 in
f2.DefaultIfEmpty()
join fun3 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 3 }
equals new {
fun3.EngineId, fun3.Function } into f3
from t3 in
f3.DefaultIfEmpty()
join fun4 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 4 }
equals new {
fun4.EngineId, fun4.Function } into f4
from t4 in
f4.DefaultIfEmpty()
join fun5 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 5 }
equals new {
fun5.EngineId, fun5.Function } into f5
from t5 in
f5.DefaultIfEmpty()
join fun6 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 6 }
equals new {
fun6.EngineId, fun6.Function } into f6
from t6 in
f6.DefaultIfEmpty()
join fun7 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 7 }
equals new {
fun7.EngineId, fun7.Function } into f7
from t7 in
f7.DefaultIfEmpty()
join fun8 in
db.EngineCharacteristics on new { EngineId =
engines.Id, Function = 8 }
equals new {
fun8.EngineId, fun8.Function } into f8
from t8 in
f8.DefaultIfEmpty()
where engines.User.UserName
== User.Identity.Name
select new
{
    Id = engines.Id,
    Fun_Mmax = t1 == null ? 0 :
t1.Value,
    Fun_Mt = t2 == null ? 0 :
t2.Value,
    Fun_Mc = t3 == null ? 0 :
t3.Value,
    Fun_Gff = t4 == null ? 0 :
t4.Value,
    Fun_Cosfi = t5 == null ?
0 : t5.Value,
}
}

```

```

        Fun_Eff = t6 == null ? 0
: t6.Value,
        Fun_M = t7 == null ? 0 :
t7.Value,
        Fun_Tw = t8 == null ? 0 :
t8.Value
    };
    foreach (var r in resfunc)
    {
        ResultsFuncModel rf = new
ResultsFuncModel
        {
            Id = r.Id,
            Fun_Mmax = r.Fun_Mmax,
            Fun_Mt = r.Fun_Mt,
            Fun_Mc = r.Fun_Mc,
            Fun_Gff = r.Fun_Gff,
            Fun_Cosfi = r.Fun_Cosfi,
            Fun_Eff = r.Fun_Eff,
            Fun_M = r.Fun_M,
            Fun_Tw = r.Fun_Tw
        };
        if (rf.Fun_Mmax == 0)
            rf.Fun_Mmax = null;
        if (rf.Fun_Mt == 0)
            rf.Fun_Mt = null;
        if (rf.Fun_Mc == 0)
            rf.Fun_Mc = null;
        if (rf.Fun_Gff == 0)
            rf.Fun_Gff = null;
        if (rf.Fun_Cosfi == 0)
            rf.Fun_Cosfi = null;
        if (rf.Fun_Eff == 0)
            rf.Fun_Eff = null;
        if (rf.Fun_M == 0)
            rf.Fun_M = null;
        if (rf.Fun_Tw == 0)
            rf.Fun_Tw = null;
        resf.Add(rf);
    }

    var res = from res1 in resa
              join res2 in resf on res1.Id
equals res2.Id
              select new
              {
                  Id = res1.Id,
                  Date = res1.Date,
                  Arg_Ds = res1.Arg_Ds,
                  Arg_Ls = res1.Arg_Ls,
                  Arg_bzs = res1.Arg_bzs,
                  Arg_bzr = res1.Arg_bzr,
                  Arg_has = res1.Arg_has,
                  Arg_har = res1.Arg_har,
                  Arg_Da = res1.Arg_Da,
                  Fun_Mmax = res2.Fun_Mmax,
                  Fun_Mt = res2.Fun_Mt,
                  Fun_Mc = res2.Fun_Mc,
                  Fun_Gff = res2.Fun_Gff,
                  Fun_Cosfi = res2.Fun_Cosfi,
                  Fun_Eff = res2.Fun_Eff,
                  Fun_M = res2.Fun_M,
                  Fun_Tw = res2.Fun_Tw
              };
    foreach (var r in res)
    {
        ResultsViewModel resengine = new
ResultsViewModel
        {
            Id = r.Id,
            Date =
r.Date.ToShortDateString(),
            Time =
r.Date.ToShortTimeString(),
            Arg_Ds = r.Arg_Ds,
            Arg_Ls = r.Arg_Ls,
            Arg_bzs = r.Arg_bzs,
            Arg_bzr = r.Arg_bzr,
            Arg_has = r.Arg_has,
            Arg_har = r.Arg_har,
            Arg_Da = r.Arg_Da,
            Fun_Mmax = r.Fun_Mmax,
            Fun_Mt = r.Fun_Mt,
            Fun_Mc = r.Fun_Mc,
            Fun_Gff = r.Fun_Gff,
            Fun_Cosfi = r.Fun_Cosfi,
            Fun_Eff = r.Fun_Eff,
            Fun_M = r.Fun_M,
            Fun_Tw = r.Fun_Tw
        };
        resengines.Add(resengine);
    }

    return View();

    public string GetData()
    {
        return
JsonConvert.SerializeObject(resengines);
    }

    public ActionResult About()
    {
        return View();
    }

    [AllowAnonymous]
    public ActionResult ChangeCulture(string
lang)
    {
        string returnUrl =
Request.UrlReferrer.AbsolutePath;
        List<string> cultures = new
List<string>() { "ru", "uk", "en" };
        if (!cultures.Contains(lang))
        {
            lang = "ru";
        }
        HttpCookie cookie =
Request.Cookies["lang"];
        if (cookie != null)
            cookie.Value = lang;
        else
        {
            cookie = new HttpCookie("lang");
            cookie.HttpOnly = false;
            cookie.Value = lang;
            cookie.Expires =
DateTime.Now.AddYears(1);
        }
        Response.Cookies.Add(cookie);
        return Redirect(returnUrl);
    }

    public enum SettingsMessageId
    {
        SaveSettingsSuccess,
        SaveSettingsError,
        NeedSetMethods,
        NeedSetLimitations,
        NeedSetCriteria
    }
}

```

```

using System;
using System.Linq;
using System.Threading.Tasks;
using System.Web;
using System.Web.Mvc;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.Owin;
using Microsoft.Owin.Security;
using Диссертация.Models;
using Диссертация.Filters;

namespace Диссертация.Controllers
{
    [Culture]
    [Authorize]
    public class ManageController : Controller
    {
        private ApplicationSignInManager
        _signInManager;
        private ApplicationUserManager _userManager;

        public ManageController()
        {
        }

        public
        ManageController(ApplicationUserManager userManager,
        ApplicationSignInManager signInManager)
        {
            UserManager = userManager;
            SignInManager = signInManager;
        }

        public ApplicationSignInManager SignInManager
        {
            get
            {
                return _signInManager ??
                HttpContext.GetOwinContext().Get<ApplicationSignInMan
                ager>();
            }
            private set
            {
                _signInManager = value;
            }
        }

        public ApplicationUserManager UserManager
        {
            get
            {
                return _userManager ??
                HttpContext.GetOwinContext().GetUserManager<Applicati
                onUserManager>();
            }
            private set
            {
                _userManager = value;
            }
        }

        //
        // GET: /Manage/Index
        public async Task<ActionResult>
        Index(ManageMessageId? message)
        {
            ViewBag.StatusMessage =
                message ==
                ManageMessageId.ChangePasswordSuccess ?
                Resources.Resource.ChangePasswordSuccess
                : message ==
                ManageMessageId.SetPasswordSuccess ?
                Resources.Resource.SetPasswordSuccess

```

```

                : message ==
                ManageMessageId.SetTwoFactorSuccess ? "Your two-
                factor authentication provider has been set."
                : message == ManageMessageId.Error ?
                Resources.Resource.Error
                : message ==
                ManageMessageId.AddPhoneSuccess ? "Your phone number
                was added."
                : message ==
                ManageMessageId.RemovePhoneSuccess ? "Your phone
                number was removed."
                : message ==
                ManageMessageId.EditAccountSuccess ?
                Resources.Resource.EditAccountSuccess
                : "";

            ApplicationUser user = await
            UserManager.FindByNameAsync(User.Identity.Name);
            var userId = User.Identity.GetUserId();
            var model = new IndexViewModel
            {
                HasPassword = HasPassword(),
                PhoneNumber = await
                UserManager.GetPhoneNumberAsync(userId),
                TwoFactor = await
                UserManager.GetTwoFactorEnabledAsync(userId),
                Logins = await
                UserManager.GetLoginsAsync(userId),
                BrowserRemembered = await
                AuthenticationManager.TwoFactorBrowserRememberedAsync
                (userId),
                DateBirth = user.DateBirth
            };

            return View(model);
        }

        //
        // POST: /Manage/RemoveLogin
        [HttpPost]
        [ValidateAntiForgeryToken]
        public async Task<ActionResult>
        RemoveLogin(string loginProvider, string providerKey)
        {
            ManageMessageId? message;
            var result = await
            UserManager.RemoveLoginAsync(User.Identity.GetUserId(
            ), new UserLoginInfo(loginProvider, providerKey));
            if (result.Succeeded)
            {
                var user = await
                UserManager.FindByIdAsync(User.Identity.GetUserId());
                if (user != null)
                {
                    await
                    SignInManager.SignInAsync(user, isPersistent: false,
                    rememberBrowser: false);
                }
                message =
                ManageMessageId.RemoveLoginSuccess;
            }
            else
            {
                message = ManageMessageId.Error;
            }
            return RedirectToAction("ManageLogins",
            new { Message = message });
        }

        //
        // GET: /Manage/AddPhoneNumber
        public ActionResult AddPhoneNumber()
        {
            return View();
        }

```

```

    }

    //
    // POST: /Manage/AddPhoneNumber
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
AddPhoneNumber(AddPhoneNumberViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        // Generate the token and send it
        var code = await
UserManager.GenerateChangePhoneNumberTokenAsync(User.
Identity.GetUserId(), model.Number);
        if (UserManager.SmsService != null)
        {
            var message = new IdentityMessage
            {
                Destination = model.Number,
                Body = "Your security code is: "
+ code
            };
            await
UserManager.SmsService.SendAsync(message);
        }
        return
RedirectToAction("VerifyPhoneNumber", new {
PhoneNumber = model.Number });
    }

    //
    // POST:
    /Manage/EnableTwoFactorAuthentication
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
EnableTwoFactorAuthentication()
    {
        await
UserManager.SetTwoFactorEnabledAsync(User.Identity.Ge
tUserId(), true);
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("MainMenu",
"Manage");
    }

    //
    // POST:
    /Manage/DisableTwoFactorAuthentication
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
DisableTwoFactorAuthentication()
    {
        await
UserManager.SetTwoFactorEnabledAsync(User.Identity.Ge
tUserId(), false);
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index",
"Manage");
    }

```

```

    }

    //
    // GET: /Manage/VerifyPhoneNumber
    public async Task<ActionResult>
VerifyPhoneNumber(string phoneNumber)
    {
        var code = await
UserManager.GenerateChangePhoneNumberTokenAsync(User.
Identity.GetUserId(), phoneNumber);
        // Send an SMS through the SMS provider
        to verify the phone number
        return phoneNumber == null ?
View("Error") : View(new VerifyPhoneNumberViewModel {
PhoneNumber = phoneNumber });
    }

    //
    // POST: /Manage/VerifyPhoneNumber
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
VerifyPhoneNumber(VerifyPhoneNumberViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        var result = await
UserManager.ChangePhoneNumberAsync(User.Identity.GetU
serId(), model.PhoneNumber, model.Code);
        if (result.Succeeded)
        {
            var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
            if (user != null)
            {
                await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
            }
            return RedirectToAction("Index", new
{ Message = ManageMessageId.AddPhoneSuccess });
        }
        // If we got this far, something failed,
        redisplay form
        ModelState.AddModelError("", "Failed to
        verify phone");
        return View(model);
    }

    //
    // POST: /Manage/RemovePhoneNumber
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
RemovePhoneNumber()
    {
        var result = await
UserManager.SetPhoneNumberAsync(User.Identity.GetUser
Id(), null);
        if (!result.Succeeded)
        {
            return RedirectToAction("Index", new
{ Message = ManageMessageId.Error });
        }
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user != null)
        {
            await SignInManager.SignInAsync(user,
isPersistent: false, rememberBrowser: false);
        }
        return RedirectToAction("Index", new {
Message = ManageMessageId.RemovePhoneSuccess });
    }

```

```

    }

    //
    // GET: /Manage/ChangePassword
    public ActionResult ChangePassword()
    {
        return View();
    }

    //
    // POST: /Manage/ChangePassword
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
ChangePassword(ChangePasswordViewModel model)
    {
        if (!ModelState.IsValid)
        {
            return View(model);
        }
        var result = await
UserManager.ChangePasswordAsync(User.Identity.GetUserId(), model.OldPassword, model.NewPassword);
        if (result.Succeeded)
        {
            var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
            if (user != null)
            {
                await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
            }
            return RedirectToAction("Index", new
{ Message = ManageMessageId.ChangePasswordSuccess });
        }
        AddErrors(result);
        return View(model);
    }

    //
    // GET: /Manage/SetPassword
    public ActionResult SetPassword()
    {
        return View();
    }

    //
    // POST: /Manage/SetPassword
    [HttpPost]
    [ValidateAntiForgeryToken]
    public async Task<ActionResult>
SetPassword(SetPasswordViewModel model)
    {
        if (ModelState.IsValid)
        {
            var result = await
UserManager.AddPasswordAsync(User.Identity.GetUserId(), model.NewPassword);
            if (result.Succeeded)
            {
                var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
                if (user != null)
                {
                    await
SignInManager.SignInAsync(user, isPersistent: false,
rememberBrowser: false);
                }
                return RedirectToAction("Index",
new { Message = ManageMessageId.SetPasswordSuccess
});
            }
            AddErrors(result);
        }
    }

```

```

        // If we got this far, something failed,
        redisplay form
        return View(model);
    }

    //
    // GET: /Manage/ManageLogins
    public async Task<ActionResult>
ManageLogins(ManageMessageId? message)
    {
        ViewBag.StatusMessage =
            message ==
ManageMessageId.RemoveLoginSuccess ? "The external
login was removed."
            : message == ManageMessageId.Error ?
            "An error has occurred."
            : "";
        var user = await
UserManager.FindByIdAsync(User.Identity.GetUserId());
        if (user == null)
        {
            return View("Error");
        }
        var userLogins = await
UserManager.GetLoginsAsync(User.Identity.GetUserId());
        ;
        var otherLogins =
AuthenticationManager.GetExternalAuthenticationTypes(
).Where(auth => userLogins.All(ul =>
auth.AuthenticationType !=
ul.LoginProvider)).ToList();
        ViewBag.ShowRemoveButton =
user.PasswordHash != null || userLogins.Count > 1;
        return View(new ManageLoginsViewModel
        {
            CurrentLogins = userLogins,
            OtherLogins = otherLogins
        });
    }

    //
    // POST: /Manage/LinkLogin
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult LinkLogin(string
provider)
    {
        // Request a redirect to the external
        login provider to link a login for the current user
        return new
AccountController.ChallengeResult(provider,
Url.Action("LinkLoginCallback", "Manage"),
User.Identity.GetUserId());
    }

    //
    // GET: /Manage/LinkLoginCallback
    public async Task<ActionResult>
LinkLoginCallback()
    {
        var loginInfo = await
AuthenticationManager.GetExternalLoginInfoAsync(XsrfKey, User.Identity.GetUserId());
        if (loginInfo == null)
        {
            return
RedirectToAction("ManageLogins", new { Message =
ManageMessageId.Error });
        }
        var result = await
UserManager.AddLoginAsync(User.Identity.GetUserId(),
loginInfo.Login);
        return result.Succeeded ?
RedirectToAction("ManageLogins") :

```

```

RedirectToAction("ManageLogins", new { Message =
ManageMessageId.Error });
}

public async Task<ActionResult> EditAccount()
{
    ApplicationUser user = await
    UserManager.FindByNameAsync(User.Identity.Name);
    if (user != null)
    {
        EditAccountModel model = new
        EditAccountModel { DateBirth = (user.DateBirth !=
        null ? user.DateBirth.Value.ToShortDateString() :
        DateTime.Today.ToShortDateString()) };
        return View(model);
    }
    return RedirectToAction("Login",
    "Account");
}

[HttpPost]
public async Task<ActionResult>
EditAccount(EditAccountModel model)
{
    if (ModelState.IsValid)
    {
        ApplicationUser user = await
        UserManager.FindByNameAsync(User.Identity.Name);
        if (user != null)
        {
            user.DateBirth =
            TryParseDatetime(model.DateBirth);
            IdentityResult result = await
            UserManager.UpdateAsync(user);
            if (result.Succeeded)
            {
                return
                RedirectToAction("Index", new { Message =
                ManageMessageId.EditAccountSuccess });
            }
            else
            {
                ModelState.AddModelError("",
                Resources.Resource.Error);
            }
        }
        else
        {
            ModelState.AddModelError("",
            Resources.Resource.UserNotFound);
        }
    }

    return View(model);
}

protected override void Dispose(bool
disposing)
{
    if (disposing && _userManager != null)
    {
        _userManager.Dispose();
        _userManager = null;
    }

    base.Dispose(disposing);
}

#region Helpers

```

```

using

```

```

// Used for XSRF protection when adding
external logins
private const string XsrfKey = "XsrfId";

private IAuthenticationManager
AuthenticationManager
{
    get
    {
        return
        HttpContext.GetOwinContext().Authentication;
    }
}

private void AddErrors(IdentityResult result)
{
    foreach (var error in result.Errors)
    {
        ModelState.AddModelError("", error);
    }
}

private bool HasPassword()
{
    var user =
    UserManager.FindById(User.Identity.GetUserId());
    if (user != null)
    {
        return user.PasswordHash != null;
    }
    return false;
}

private bool HasPhoneNumber()
{
    var user =
    UserManager.FindById(User.Identity.GetUserId());
    if (user != null)
    {
        return user.PhoneNumber != null;
    }
    return false;
}

private DateTime? TryParseDatetime(string
value)
{
    DateTime db;
    DateTime.TryParse(value, out db);
    if (db >= new DateTime(1900, 1, 1))
        return db;
    else
        return null;
}

public enum ManageMessageId
{
    AddPhoneSuccess,
    ChangePasswordSuccess,
    SetTwoFactorSuccess,
    SetPasswordSuccess,
    RemoveLoginSuccess,
    RemovePhoneSuccess,
    Error,
    EditAccountSuccess
}

#endregion
}

```

```

System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Globalization;
using System.Threading;
using System.Web.Mvc;

namespace Диссертация.Filters
{
    public class CultureAttribute : FilterAttribute,
    IActionFilter
    {
        public void
        OnActionExecuted(ActionExecutedContext filterContext)
        {
            //не реализован
        }

        public void
        OnActionExecuting(ActionExecutingContext
        filterContext)
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using Диссертация.Annotations;

namespace Диссертация.Models
{
    public class ExternalLoginConfirmationViewModel
    {
        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }

    public class ExternalLoginListViewModel
    {
        public string ReturnUrl { get; set; }
    }

    public class SendCodeViewModel
    {
        public string SelectedProvider { get; set; }
        public
        ICollection<System.Web.Mvc.SelectListItem> Providers
        { get; set; }
        public string ReturnUrl { get; set; }
        public bool RememberMe { get; set; }
    }

    public class VerifyCodeViewModel
    {
        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        public string Provider { get; set; }

        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [Display(Name = "Code")]
        public string Code { get; set; }
        public string ReturnUrl { get; set; }

        [Display(Name = "Remember this browser?")]
        public bool RememberBrowser { get; set; }
    }
}

```

```

{
    string cultureName = null;
    HttpCookie cultureCookie =
    filterContext.HttpContext.Request.Cookies["lang"];
    if (cultureCookie != null)
        cultureName = cultureCookie.Value;
    else
        cultureName = "ru";

    List<string> cultures = new
    List<string>() { "ru", "uk", "en" };
    if (!cultures.Contains(cultureName))
    {
        cultureName = "ru";
    }
    Thread.CurrentThread.CurrentCulture =
    CultureInfo.CreateSpecificCulture(cultureName);
    Thread.CurrentThread.CurrentUICulture =
    CultureInfo.CreateSpecificCulture(cultureName);
}
}
}

```

```

        public bool RememberMe { get; set; }
    }

    public class ForgotViewModel
    {
        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }

    public class LoginViewModel
    {
        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [Display(Name =
        nameof(Resources.Resource.Login),
        ResourceType=typeof(Resources.Resource))]
        public string UserName { get; set; }

        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [DataType(DataType.Password)]
        [Display(Name =
        nameof(Resources.Resource.Password), ResourceType =
        typeof(Resources.Resource))]
        public string Password { get; set; }

        [Display(Name =
        nameof(Resources.Resource.RememberMe), ResourceType =
        typeof(Resources.Resource))]
        public bool RememberMe { get; set; }
    }

    public class RegisterViewModel
    {
        [Required(ErrorMessageResourceName =
        nameof(Resources.Resource.FieldRequired),
        ErrorMessageResourceType =
        typeof(Resources.Resource))]
        [Display(Name =
        nameof(Resources.Resource.Login), ResourceType =
        typeof(Resources.Resource))]
    }
}

```

```

        public string UserName { get; set; }

        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }

        [DisplayFormat(NullDisplayText = "")]
        [Display(Name =
nameof(Resources.Resource.DateBirth), ResourceType =
typeof(Resources.Resource))]
        public string DateBirth { get; set; }

        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [StringLength(100, ErrorMessageResourceName =
nameof(Resources.Resource.StringMinimumLength),
ErrorMessageResourceType =
typeof(Resources.Resource), MinimumLength = 8)]
        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.Password), ResourceType =
typeof(Resources.Resource))]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.PasswordConfirmation),
ResourceType = typeof(Resources.Resource))]
        [Compare("Password", ErrorMessageResourceName
= nameof(Resources.Resource.DifferentPasswords),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        public string ConfirmPassword { get; set; }
    }

    public class ResetPasswordViewModel
    {
        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),

```

```

ErrorMessageResourceType =
typeof(Resources.Resource))]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }

        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [StringLength(100, ErrorMessageResourceName =
nameof(Resources.Resource.StringMinimumLength),
ErrorMessageResourceType =
typeof(Resources.Resource), MinimumLength = 8)]
        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.Password), ResourceType =
typeof(Resources.Resource))]
        public string Password { get; set; }

        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.PasswordConfirmation),
ResourceType = typeof(Resources.Resource))]
        [Compare("Password", ErrorMessageResourceName
= nameof(Resources.Resource.DifferentPasswords),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        public string ConfirmPassword { get; set; }

        public string Code { get; set; }
    }

    public class ForgotPasswordViewModel
    {
        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [EmailAddress]
        [Display(Name = "Email")]
        public string Email { get; set; }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

```

```

namespace Диссертация.Models

```

```

{
    public class Language
    {
        public int Id { get; set; }
        public int LCID { get; set; }
        public string Name { get; set; }
    }

    public class Method
    {
        public int Id { get; set; }
        public int MethodType { get; set; }
        public int LanguageId { get; set; }
        public Language Language { get; set; }
        public string Name { get; set; }
    }

    public class MethodSetting
    {
        public int Id { get; set; }
    }
}

```

```

        public bool Usage { get; set; }
        public string UserId { get; set; }
        public ApplicationUser User { get; set; }
        public int MethodType { get; set; }
    }
}

```

```

public class Function
{
    public int Id { get; set; }
    public int FunctionType { get; set; }
    public int LanguageId { get; set; }
    public Language Language { get; set; }
    public string NameLong { get; set; }
    public string NameShort { get; set; }
}

public class FunctionValue
{
    public int Id { get; set; }
    public int Sign { get; set; }
    public double DefaultValue { get; set; }
    public double MinValue { get; set; }
    public double MaxValue { get; set; }
}

public class FunctionSetting

```

```

{
    public int Id { get; set; }
    public bool Limitation { get; set; }
    public bool Criteria { get; set; }
    public string UserId { get; set; }
    public ApplicationUser User { get; set; }
    public int FunctionType { get; set; }
}

public class Argument
{
    public int Id { get; set; }
    public int AgrumentType { get; set; }
    public int LanguageId { get; set; }
    public Language Language { get; set; }
    public string NameLong { get; set; }
    public string NameShort { get; set; }
}

public class Engine
{
    public int Id { get; set; }
    public DateTime EngineDate { get; set; }
    public int Method { get; set; }
}

```

```

    public string UserId { get; set; }
    public ApplicationUser User { get; set; }
}

public class EngineParameter
{
    public int Id { get; set; }
    public int EngineId { get; set; }
    public Engine Engine { get; set; }
    public int Argument { get; set; }
    public double Value { get; set; }
}

public class EngineCharacteristic
{
    public int Id { get; set; }
    public int EngineId { get; set; }
    public Engine Engine { get; set; }
    public int Function { get; set; }
    public bool Criteria { get; set; }
    public double Value { get; set; }
}
}

```

```

using System;
using System.Data.Entity;
using System.Security.Claims;
using System.Threading.Tasks;
using Microsoft.AspNet.Identity;
using Microsoft.AspNet.Identity.EntityFramework;

namespace Диссертация.Models
{
    public class ApplicationUser : IdentityUser
    {
        public DateTime? DateBirth { get; set; }

        public async Task<ClaimsIdentity>
GenerateUserIdentityAsync(UserManager<ApplicationUser>
> manager)
        {
            var userIdentity = await
manager.CreateIdentityAsync(this,
DefaultAuthenticationTypes.ApplicationCookie);
            return userIdentity;
        }
    }

    public class ApplicationDbContext :
IdentityDbContext<ApplicationUser>
    {
        public ApplicationDbContext()

```

```

        : base("DefaultConnection",
throwIfV1Schema: false)
        {
        }

        public static ApplicationDbContext Create()
        {
            return new ApplicationDbContext();
        }

        public DbSet<Argument> Arguments { get; set; }
        public DbSet<Language> Languages { get; set; }
        public DbSet<Method> Methods { get; set; }
        public DbSet<MethodSetting> MethodSettings {
get; set; }
        public DbSet<Function> Functions { get; set; }
        public DbSet<FunctionValue> FunctionValues {
get; set; }
        public DbSet<FunctionSetting>
FunctionSettings { get; set; }
        public DbSet<Engine> Engines { get; set; }
        public DbSet<EngineParameter>
EngineParameters { get; set; }
        public DbSet<EngineCharacteristic>
EngineCharacteristics { get; set; }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace Диссертация.Models
{
    public class MenuItem
    {
        public MenuItem() { }
        public int Id { get; set; }
        public string Text { get; set; }
        public string Definition { get; set; }
    }
}

```

```

        public string Value { get; set; }
        public int Sign { get; set; }
        public double MinValue { get; set; }
        public double MaxValue { get; set; }
    }

    public class MainMenuViewModel
    {
        public IList<MenuItem> Limitation { get;
set; }
        public IList<int> LimitationId { get; set; }
        public IList<double> LimitationValue { get;
set; }
    }
}

```

```

        public IList<MenuItem> Argument { get;
set; }
        public string ResultText { get; set; }

        public MainMenuViewModel()
        {
            Limitation = new List<MenuItem>();

```

```

            LimitationId = new List<int>();
            LimitationValue = new List<double>();
            Argument = new List<MenuItem>();
        }
    }
}

```

```

using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using Microsoft.AspNet.Identity;
using Microsoft.Owin.Security;
using Диссертация.Annotations;

namespace Диссертация.Models
{
    public class IndexViewModel
    {
        public bool HasPassword { get; set; }
        public IList<UserLoginInfo> Logins { get;
set; }
        public string PhoneNumber { get; set; }
        public bool TwoFactor { get; set; }
        public bool BrowserRemembered { get; set; }
        public DateTime? DateBirth { get; set; }
    }

    public class ManageLoginsViewModel
    {
        public IList<UserLoginInfo> CurrentLogins {
get; set; }
        public IList<AuthenticationDescription>
OtherLogins { get; set; }
    }

    public class FactorViewModel
    {
        public string Purpose { get; set; }
    }

    public class SetPasswordViewModel
    {
        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [StringLength(100, ErrorMessageResourceName =
nameof(Resources.Resource.StringMinimumLength),
ErrorMessageResourceType =
typeof(Resources.Resource), MinimumLength = 8)]
        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.NewPassword), ResourceType
= typeof(Resources.Resource))]
        public string NewPassword { get; set; }

        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.PasswordConfirmation),
ResourceType = typeof(Resources.Resource))]
        [Compare("Password", ErrorMessageResourceName
= nameof(Resources.Resource.DifferentPasswords),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        public string ConfirmPassword { get; set; }
    }

    public class ChangePasswordViewModel
    {
        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),

```

```

ErrorMessageResourceType =
typeof(Resources.Resource))]
        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.CurrentPassword),
ResourceType = typeof(Resources.Resource))]
        public string OldPassword { get; set; }

        [Required(ErrorMessageResourceName =
nameof(Resources.Resource.FieldRequired),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        [StringLength(100, ErrorMessageResourceName =
nameof(Resources.Resource.StringMinimumLength),
ErrorMessageResourceType =
typeof(Resources.Resource), MinimumLength = 8)]
        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.NewPassword), ResourceType
= typeof(Resources.Resource))]
        public string NewPassword { get; set; }

        [DataType(DataType.Password)]
        [Display(Name =
nameof(Resources.Resource.NewPasswordConfirmation),
ResourceType = typeof(Resources.Resource))]
        [Compare("NewPassword",
ErrorMessageResourceName =
nameof(Resources.Resource.DifferentPasswords),
ErrorMessageResourceType =
typeof(Resources.Resource))]
        public string ConfirmPassword { get; set; }
    }

    public class AddPhoneNumberViewModel
    {
        [Required]
        [Phone]
        [Display(Name = "Phone Number")]
        public string Number { get; set; }
    }

    public class VerifyPhoneNumberViewModel
    {
        [Required]
        [Display(Name = "Code")]
        public string Code { get; set; }

        [Required]
        [Phone]
        [Display(Name = "Phone Number")]
        public string PhoneNumber { get; set; }
    }

    public class ConfigureTwoFactorViewModel
    {
        public string SelectedProvider { get; set; }
        public
ICollection<System.Web.Mvc.SelectListItem> Providers
{ get; set; }
    }

    public class EditAccountModel
    {

```



```

[DisplayFormat(NullDisplayText = "")]
[Display(Name =
nameof(Resources.Resource.DateBirth), ResourceType =
typeof(Resources.Resource))]]

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

```

```

namespace Диссертация.Models
{
    public class ResultsArgsModel
    {
        public int Id { get; set; }
        public DateTime Date { get; set; }
        public double Arg_Ds { get; set; }
        public double Arg_Ls { get; set; }
        public double Arg_bzs { get; set; }
        public double Arg_bzr { get; set; }
        public double Arg_has { get; set; }
        public double Arg_har { get; set; }
        public double Arg_Da { get; set; }
    }
    public class ResultsFuncModel
    {
        public int Id { get; set; }
        public double? Fun_Mmax { get; set; }
        public double? Fun_Mt { get; set; }
        public double? Fun_Mc { get; set; }
        public double? Fun_Gff { get; set; }
        public double? Fun_Cosfi { get; set; }
    }
}

```

```

public string DateBirth { get; set; }
}

```

```

public double? Fun_Eff { get; set; }
public double? Fun_M { get; set; }
public double? Fun_Tw { get; set; }
}

```

```

public class ResultsViewModel
{
    public int Id { get; set; }
    public string Date { get; set; }
    public string Time { get; set; }
    public double Arg_Ds { get; set; }
    public double Arg_Ls { get; set; }
    public double Arg_bzs { get; set; }
    public double Arg_bzr { get; set; }
    public double Arg_has { get; set; }
    public double Arg_har { get; set; }
    public double Arg_Da { get; set; }
    public double? Fun_Mmax { get; set; }
    public double? Fun_Mt { get; set; }
    public double? Fun_Mc { get; set; }
    public double? Fun_Gff { get; set; }
    public double? Fun_Cosfi { get; set; }
    public double? Fun_Eff { get; set; }
    public double? Fun_M { get; set; }
    public double? Fun_Tw { get; set; }
}

```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Data.Entity;
using System.Web.Mvc;

```

```

namespace Диссертация.Models
{
    public class SettingsViewModel
    {
        public IList<int> SelectedMethods { get; set; }
    }
    public IList<SelectListItem> MethodsList {
get; set; }
    public IList<int> SelectedLimitations { get;
set; }
    public IList<SelectListItem> LimitationsList
{ get; set; }
}

```

```

public IList<int> SelectedCriteriaas { get;
set; }
    public IList<SelectListItem> CriteriaasList {
get; set; }

    public SettingsViewModel()
    {
        SelectedMethods = new List<int>();
        MethodsList = new List<SelectListItem>();
        SelectedLimitations = new List<int>();
        LimitationsList = new
List<SelectListItem>();
        SelectedCriteriaas = new List<int>();
        CriteriaasList = new
List<SelectListItem>();
    }
}

```

```

@model Диссертация.Models.ExternalLoginListViewModel
@using Microsoft.Owin.Security

```

```

<h4>Use another service to log in.</h4>

```

```

<hr />

```

```

@{
    var loginProviders =
Context.GetOwinContext().Authentication.GetExternalAu
thenticationTypes();
    if (loginProviders.Count() == 0) {
        <div>
            <p>
                There are no external authentication
                services configured. See <a

```

```

href="http://go.microsoft.com/fwlink/?LinkId=403804">
this article</a>

```

```

        for details on setting up this
        ASP.NET application to support logging in via
        external services.
    </p>

```

```

</div>
}
else {
    using (Html.BeginForm("ExternalLogin",
"Account", new { ReturnUrl = Model.ReturnUrl })) {
        @Html.AntiForgeryToken()
        <div id="socialLoginList">
            <p>

```

```

        @foreach
(AuthenticationDescription p in loginProviders) {
            <button type="submit"
class="btn btn-default" id="@p.AuthenticationType"
name="provider" value="@p.AuthenticationType"
title="Log in using your @p.Caption
account">@p.AuthenticationType</button>

```

```

        }
    }
</div>
</p>
}

```

```

@{
    ViewBag.Title = Resources.Resource.ConfirmEmail;
}

<h2>@ViewBag.Title</h2>
<div>
    <p>

```

```

        @Resources.Resource.ConfirmEmailThanks <a
href="/EngineOpt/Account/Login" id="loginLink"
class="btn-
link">@Resources.Resource.ConfirmEmailPress</a>
    </p>
</div>

```

```

@{
    ViewBag.Title =
Resources.Resource.DisplayEmailHeader;
}

```

```

<h2>@ViewBag.Title</h2>

<p>
    @Resources.Resource.DisplayEmailBody
</p>

```

```

@model Дипсертация.Models.ExternalLoginConfirmationViewModel
@{
    ViewBag.Title = "Register";
}

<h2>@ViewBag.Title.</h2>
<h3>Associate your @ViewBag.LoginProvider
account.</h3>

@using (Html.BeginForm("ExternalLoginConfirmation",
"Account", new { ReturnUrl = ViewBag.ReturnUrl },
FormMethod.Post, new { @class = "form-horizontal",
role = "form" })))
{
    @Html.AntiForgeryToken()

    <h4>Association Form</h4>
    <hr />
    @Html.ValidationSummary(true, "", new { @class =
"text-danger" })
    <p class="text-info">
        You've successfully authenticated with
    <strong>@ViewBag.LoginProvider</strong>.
    Please enter a user name for this site below
    and click the Register button to finish

```

```

        logging in.
    </p>
    <div class="form-group">
        @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Email, new {
@class = "form-control" })
            @Html.ValidationMessageFor(m => m.Email,
"", new { @class = "text-danger" })
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" class="btn btn-
default" value="Register" />
            </div>
        </div>
    }

    @section Scripts {
        @Scripts.Render("~/bundles/jqueryval")
    }

```

```

@{
    ViewBag.Title = "Login Failure";
}

<hgroup>

```

```

    <h2>@ViewBag.Title.</h2>
    <h3 class="text-danger">Unsuccessful login with
service.</h3>
</hgroup>

```

```

@model Дипсертация.Models.ForgotPasswordViewModel
@{
    ViewBag.Title =
Resources.Resource.ForgotPasswordHeader;
}

<h2>@ViewBag.Title</h2>

```

```

@using (Html.BeginForm("ForgotPassword", "Account",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" })))
{
    @Html.AntiForgeryToken()
    <h4>@Resources.Resource.EnterYourEmail.</h4>
    <hr />
    @Html.ValidationSummary("", new { @class = "text-
danger" })

```

```

        <div class="form-group">
            @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
            <div class="col-md-10">
                @Html.TextBoxFor(m => m.Email, new {
@class = "form-control" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">

```

```

@{
    ViewBag.Title =
Resources.Resource.ForgotPasswordConfirmationHeader;
}

<hgroup class="title">
    <h1>@ViewBag.Title</h1>

```

```

@using Диссертация.Models
@model LoginViewModel
@{
    ViewBag.Title = @Resources.Resource.LoginHeader;
}

<h2>@ViewBag.Title</h2>
<div class="row">
    <div class="col-md-8">
        <section id="loginForm">
            @using (Html.BeginForm("Login",
"Account", new { returnUrl = ViewBag.ReturnUrl },
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))
            {
                @Html.AntiForgeryToken()
                <hr />
                @Html.ValidationSummary(true, "", new
{ @class = "text-danger" })
                <div class="form-group">
                    @Html.LabelFor(m => m.UserName,
new { @class = "col-md-2 control-label" })
                    <div class="col-md-10">
                        @Html.TextBoxFor(m =>
m.UserName, new { @class = "form-control" })
                        @Html.ValidationMessageFor(m
=> m.UserName, "", new { @class = "text-danger" })
                    </div>
                </div>
                <div class="form-group">
                    @Html.LabelFor(m => m.Password,
new { @class = "col-md-2 control-label" })
                    <div class="col-md-10">
                        @Html.PasswordFor(m =>
m.Password, new { @class = "form-control" })
                        @Html.ValidationMessageFor(m
=> m.Password, "", new { @class = "text-danger" })
                    </div>

```

```

@model Диссертация.Models.RegisterViewModel
@{
    ViewBag.Title = Resources.Resource.Register;
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("Register", "Account",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))

```

```

        <input type="submit" class="btn btn-
primary" value="@Resources.Resource.Send" />
    </div>
</div>
}

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

</hgroup>
<div>
    <p>
        @Resources.Resource.ForgotPasswordConfirmationBody.
    </p>
</div>

```

```

</div>
<div class="form-group">
    <div class="col-md-offset-2 col-
md-10">
        <div class="checkbox">
            @Html.CheckBoxFor(m =>
m.RememberMe)
            @Html.LabelFor(m =>
m.RememberMe)
        </div>
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-
md-10">
        <input type="submit"
value="@Resources.Resource.Enter" class="btn btn-
primary" />
    </div>
</div>
<p>
    <a
href="/EngineOpt/Account/Register" class="btn-
link">@Resources.Resource.RegisterNewUser</a>
</p>
<p>
    <a
href="/EngineOpt/Account/ForgotPassword" class="btn-
link">@Resources.Resource.ForgotPasswordHeader</a>
</p>
}
</section>
</div>
</div>

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

{
    @Html.AntiForgeryToken()
    <hr />
    @Html.ValidationSummary("", new { @class = "text-
danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.UserName, new { @class
= "col-md-2 control-label" })
        <div class="col-md-10">

```

```

        @Html.TextBoxFor(m => m.UserName, new {
@class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
    <div class="col-md-10">
        @Html.TextBoxFor(m => m.Email, new {
@class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.Password, new { @class
= "col-md-2 control-label" })
    <div class="col-md-10">
        @Html.PasswordFor(m => m.Password, new {
@class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.ConfirmPassword, new {
@class = "col-md-2 control-label" })
    <div class="col-md-10">

```

```

        @Html.PasswordFor(m => m.ConfirmPassword,
new { @class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.DateBirth, new { @class
= "col-md-2 control-label" })
    <div class="col-md-10">
        @Html.TextBoxFor(m => m.DateBirth, new {
@class = "form-control" })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" class="btn btn-
primary" value="@Resources.Resource.ToRegister" />
    </div>
</div>
}

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

@model Дипсертация.Models.ResetPasswordViewModel
@{
    ViewBag.Title = Resources.Resource.ResetPassword;
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("ResetPassword", "Account",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))
{
    @Html.AntiForgeryToken()
    <hr />
    @Html.ValidationSummary("", new { @class = "text-
danger" })
    @Html.HiddenFor(model => model.Code)
    <div class="form-group">
        @Html.LabelFor(m => m.Email, new { @class =
"col-md-2 control-label" })
        <div class="col-md-10">
            @Html.TextBoxFor(m => m.Email, new {
@class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.Password, new { @class
= "col-md-2 control-label" })

```

```

    <div class="col-md-10">
        @Html.PasswordFor(m => m.Password, new {
@class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.ConfirmPassword, new {
@class = "col-md-2 control-label" })
    <div class="col-md-10">
        @Html.PasswordFor(m => m.ConfirmPassword,
new { @class = "form-control" })
    </div>
</div>
<div class="form-group">
    <div class="col-md-offset-2 col-md-10">
        <input type="submit" class="btn btn-
primary" value="@Resources.Resource.Reset" />
    </div>
</div>
}

```

```

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

@{
    ViewBag.Title =
Resources.Resource.ResetPasswordConfirmationHeader;
}

<hgroup class="title">
    <h1>@ViewBag.Title</h1>
</hgroup>

```

```

<div>
    <p>
        @Resources.Resource.ResetPasswordThanks <a
href="/EngineOpt/Account/Login" id="loginLink"
class="btn-
link">@Resources.Resource.ConfirmEmailPress</a>
    </p>
</div>

```

```

@model Дипсертация.Models.SendCodeViewModel
@{
    ViewBag.Title = "Send";
}

<h2>@ViewBag.Title.</h2>

```

```

@using (Html.BeginForm("SendCode", "Account", new {
ReturnUrl = Model.ReturnUrl }, FormMethod.Post, new {
@class = "form-horizontal", role = "form" })) {
    @Html.AntiForgeryToken()
    @Html.Hidden("rememberMe", @Model.RememberMe)
    <h4>Send verification code</h4>
    <hr />

```

```

<div class="row">
  <div class="col-md-8">
    Select Two-Factor Authentication
  Provider:
    @Html.DropDownListFor(model =>
model.SelectedProvider, Model.Providers)
    <input type="submit" value="Submit"
class="btn btn-default" />
  </div>

```

```

@model Диссертация.Models.VerifyCodeViewModel
@{
  ViewBag.Title = "Verify";
}

<h2>@ViewBag.Title.</h2>

@using (Html.BeginForm("VerifyCode", "Account", new {
ReturnUrl = Model.ReturnUrl }, FormMethod.Post, new {
@class = "form-horizontal", role = "form" })) {
  @Html.AntiForgeryToken()
  @Html.Hidden("provider", @Model.Provider)
  @Html.Hidden("rememberMe", @Model.RememberMe)
  <h4>Enter verification code</h4>
  <hr />
  @Html.ValidationSummary("", new { @class = "text-
danger" })
  <div class="form-group">
    @Html.LabelFor(m => m.Code, new { @class =
"col-md-2 control-label" })
    <div class="col-md-10">
      @Html.TextBoxFor(m => m.Code, new {
@class = "form-control" })

```

```

</div>
}
@section Scripts {
  @Scripts.Render("~/bundles/jqueryval")
}

```

```

</div>
</div>
<div class="form-group">
  <div class="col-md-offset-2 col-md-10">
    <div class="checkbox">
      @Html.CheckBoxFor(m =>
m.RememberBrowser)
      @Html.LabelFor(m =>
m.RememberBrowser)
    </div>
  </div>
  <div class="form-group">
    <div class="col-md-offset-2 col-md-10">
      <input type="submit" class="btn btn-
default" value="Submit" />
    </div>
  </div>
}
@section Scripts {
  @Scripts.Render("~/bundles/jqueryval")
}

```

```

@{
  ViewBag.Title = Resources.Resource.Error500;
}

```

```

<h2 class="text-danger">@ViewBag.Title</h2>

```

```

@{
  ViewBag.Title = Resources.Resource.Error404;
}

```

```

<h2 class="text-danger">@ViewBag.Title</h2>

```

```

@{
  ViewBag.Title = Resources.Resource.Error401;
}

```

```

<h2 class="text-danger">@ViewBag.Title</h2>

```

```

@{
  ViewBag.Title = Resources.Resource.About;
}
<h2>@ViewBag.Title</h2>

<div>
  <p><b>EngineOpt</b>
  @Resources.Resource.AboutHeader1.</p>
  <br/>
  <p><b>EngineOpt
  @Resources.Resource.AboutHeader2:</b></p>
  <p>- @Resources.Resource.AboutStr21;</p>

```

```

<p>- @Resources.Resource.AboutStr22;</p>
<p>- @Resources.Resource.AboutStr23.</p>
<br/>
<p><b>@Resources.Resource.AboutHeader3:</b></p>
<p>1) @Resources.Resource.AboutStr31;</p>
<p>2) @Resources.Resource.AboutStr32:</p>
<p>&nbsp;- @Resources.Resource.AboutStr321;</p>
<p>&nbsp;- @Resources.Resource.AboutStr322;</p>
<p>&nbsp;- @Resources.Resource.AboutStr323;</p>
<p>3) @Resources.Resource.AboutStr33;</p>
<p>4) @Resources.Resource.AboutStr34.</p>
</div>

```

```

@model Диссертация.Models.MainMenuViewModel
@{
  ViewBag.Title = Resources.Resource.MainMenu;
}

```

```

}
<style>
  #center {

```

```

        border: 1px solid black;
        padding: 10px;
    }
    #right {
        border: 1px solid black;
        padding: 10px;
        text-align: right;
        width: 90px;
    }
    #noborder{
        border: 0;
        width: 150px;
    }
    #fixedtable{
        table-layout: fixed;
        width: 270px;
    }
    #area{
        margin-left: 15px;
        height: 300px;
        resize: none;
    }
    @@media screen and (max-device-width:480px){
        #area{
            max-width:300px;
            width: 300px;
        }
    }
    @@media screen and (min-width:481px){
        #area{
            max-width:550px;
            width: 550px;
        }
    }
}
</style>

<script type="text/javascript">
function Validate()
{
    var container, inputs;
    container =
document.getElementById('LimitationList');
    inputs = container.getElementsByTagName('input')
    for (var i = 0; i < inputs.length; i++) {
        inputs[i].value =
inputs[i].value.replace(/[\,]+/g, ',');
        if (!inputs[i].value) {

alert("@Resources.Resource.AlertLimValue");
            return false;
        }
        else if
(!$.isNumeric(inputs[i].value.replace(/[\,]+/g, ',')))
{
alert("@Resources.Resource.AlertIsNumeric");
            return false;
        }
        else if
((parseFloat(inputs[i].value.replace(/[\,]+/g, ',')) <
parseFloat(inputs[i].min.replace(/[\,]+/g, ',')))
||
(parseFloat(inputs[i].value.replace(/[\,]+/g, ',')) >
parseFloat(inputs[i].max.replace(/[\,]+/g, ','))) {
            alert("@Resources.Resource.WrongRange");
            return false;
        }
    }
    return true;
}
</script>

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("MainMenu", "Home"))

```

```

{
    <div class="container">
        <div class="row">
            <div class="col-md-3"
id="LimitationList">
                <form>
                    <fieldset>

<legend>@Resources.Resource.Limitations</legend>
                    <table id="fixedtable">
                        @foreach (var lim in
Model.Limitation)
                            {
                                <tr>
                                    <td id="right">
                                        @lim.Text
                                        @if (lim.Sign
== 1)
                                            { <text>
&#8804 </text> }
                                        else
                                            { <text>
&#8805 </text> }
                                    </td>
                                    <td id="center">
                                        <input
id="noborder" name="LimitationValue"
min="@lim.MinValue" max="@lim.MaxValue"

value="@lim.Value" placeholder="@lim.MinValue -
@lim.MaxValue" />
                                        <input
type="hidden" name="LimitationId" value="@lim.Id" />
                                    </td>
                                </tr>
                            }
                        </table>
                    </fieldset>
                </form>
            </div>
            <div class="col-md-3" id="ArgumentList">
                <form>
                    <fieldset>

<legend>@Resources.Resource.OptValues</legend>
                    <table id="fixedtable">
                        @foreach (var arg in
Model.Argument)
                            {
                                <tr>
                                    <td id="right">
                                        @arg.Text =
                                    </td>
                                    <td id="center">
                                        <input
id="noborder" name="@arg.Id" value="@arg.Value"
readonly />
                                    </td>
                                </tr>
                            }
                        </table>
                    </fieldset>
                </form>
            </div>
            <div class="col-md-6">
                <form>
                    <fieldset>

<legend>@Resources.Resource.Results</legend>
                    <div class="row">
                        <textarea id="area"
name="Result"
readonly="readonly">@Model.ResultText</textarea>
                    </div>
                </fieldset>
            </div>
        </div>
    </div>
}

```

```

        </form>
    </div>
</div>
<br />
<div class="row">
    <div class="form-group text-center">
        <input type="submit" class="btn btn-
primary" value="@Resources.Resource.Optimize"
onclick="return Validate();" />
    </div>
</div>
</div>
<hr />

```

```

@{
    ViewBag.Title = Resources.Resource.Results;
}

@section styles
{
<link href="~/Content/themes/base/jquery-ui.min.css"
rel="stylesheet" type="text/css" />
<link href="~/Content/jquery.jqGrid/ui.jqgrid.css"
rel="stylesheet" type="text/css" />

<style>
    .ui-jqgrid .ui-pg-input {
        height: 20px;
    }

    .ui-jqgrid .ui-jqgrid-pager {
        height: 40px;
    }

    .ui-jqgrid .ui-jqgrid-pager .ui-pg-div {
        font-size: 14px;
    }

    .ui-state-highlight,
    .ui-widget-content .ui-state-highlight,
    .ui-widget-header .ui-state-highlight {
        border: 1px solid #964c68;
        background: #943357;
        color: #ffffff;
    }

    .ui-state-checked {
        border: 1px solid #964c68;
        background: #943357;
    }

    .ui-state-active,
    .ui-widget-content .ui-state-active,
    .ui-widget-header .ui-state-active,
    a.ui-button:active,
    .ui-button:active,
    .ui-button.ui-state-active:hover {
        border: 1px solid #964c68;
        background: #943357;
        color: #ffffff;
    }

    .ui-icon-background,
    .ui-state-active .ui-icon-background {
        border: #964c68;
        background-color: #943357;
    }
</style>
}

@section scripts

```

```

<div>
    <p>
        @foreach (var lim in Model.Limitation)
        {
            <text>@lim.Text - @lim.Definition</text>
        }
    </p>
    <p>
        @foreach (var arg in Model.Argument)
        {
            <text>@arg.Text - @arg.Definition</text>
        }
    </p>
</div>

```

```

{
<script src="~/Scripts/jquery-ui-1.12.1.min.js"
type="text/javascript"></script>
<script src="~/Scripts/i18n/grid.locale-en.js"
type="text/javascript"></script>
<script src="~/Scripts/jquery.jqGrid.min.js"
type="text/javascript"></script>

<script type="text/javascript">
    $(document).ready(function () {
        $("#jqg").jqGrid({
            url: '@Url.Action("GetData")',
            datatype: "json",
            colNames: [ '@Resources.Resource.Date',
                '@Resources.Resource.Time',
                '@Resources.Resource.Arg_Ds',
                '@Resources.Resource.Arg_Ls',
                '@Resources.Resource.Arg_bzs',
                '@Resources.Resource.Arg_bzr',
                '@Resources.Resource.Arg_has',
                '@Resources.Resource.Arg_has',
                '@Resources.Resource.Arg_Da',
                '@Resources.Resource.Fun_Mmax',
                '@Resources.Resource.Fun_Mt',
                '@Resources.Resource.Fun_Mc',
                '@Resources.Resource.Fun_Gff',
                '@Resources.Resource.Fun_Cosfi',
                '@Resources.Resource.Fun_Eff',
                '@Resources.Resource.Fun_M',
                '@Resources.Resource.Fun_Tw' ],
            colModel: [
                { name: 'Date', index: 'Date', width: 70,
                sortable: true },
                { name: 'Time', index: 'Time', width: 60,
                sortable: true },
                { name: 'Arg_Ds', index: 'Arg_Ds', width:
                160, sortable: true },
                { name: 'Arg_Ls', index: 'Arg_Ls', width:
                160, sortable: true },
                { name: 'Arg_bzs', index: 'Arg_bzs',
                width: 150, sortable: true },
                { name: 'Arg_bzr', index: 'Arg_bzr',
                width: 150, sortable: true },
                { name: 'Arg_has', index: 'Arg_has',
                width: 150, sortable: true },
                { name: 'Arg_har', index: 'Arg_har',
                width: 150, sortable: true },
                { name: 'Arg_Da', index: 'Arg_Da', width:
                190, sortable: true },
                { name: 'Fun_Mmax', index: 'Fun_Mmax',
                width: 200, sortable: true },
                { name: 'Fun_Mt', index: 'Fun_Mt', width:
                230, sortable: true },
                { name: 'Fun_Mc', index: 'Fun_Mc', width:
                220, sortable: true },
                { name: 'Fun_Gff', index: 'Fun_Gff',
                width: 180, sortable: true },
            ]
        });
    });
}

```



```

        { name: 'Fun_Cosfi', index: 'Fun_Cosfi',
width: 150, sortable: true },
        { name: 'Fun_Eff', index: 'Fun_Eff',
width: 190, sortable: true },
        { name: 'Fun_M', index: 'Fun_M', width:
160, sortable: true },
        { name: 'Fun_Tw', index: 'Fun_Tw', width:
250, sortable: true },
    ],
    autowidth: true,
    shrinkToFit: false,
    height: 400,
    rowNum: 10,
    rowList: [10, 20, 30],
    pager: '#jpager',
    loadonce: true,
    sortname: '@Resources.Resource.Date',
    sortorder: "desc",
    hidegrid: false,
    caption: "@Resources.Resource.Engines",
    emptyrecords:
"@Resources.Resource.NoRecords",
    loadtext: "@Resources.Resource.Loading",
    pgtext: "@Resources.Resource.Page"
    });
$("#jqg").jqGrid('navGrid', '#jpager', {
    add: false,
    del: false,
    edit: false,
    view: true,
    viewtext: "@Resources.Resource.Watch",
    viewtitle:
"@Resources.Resource.ViewTitle",
    search: true,
    searchtext: "@Resources.Resource.Search",
    searchtitle:
"@Resources.Resource.SearchTitle",
    refresh: true,
    refreshtext:
"@Resources.Resource.Refresh",
    refreshtitle:
"@Resources.Resource.RefreshTitle",

```

```

        alertcap:
"@Resources.Resource.WarningCaption",
        alerttext:
"@Resources.Resource.WarningText"
    }, {}, {}, {}, {
        caption: "@Resources.Resource.Search...",
        Find: "@Resources.Resource.Find",
        Reset: "@Resources.Resource.Reset",
        multipleSearch: true,
        odata: [{ oper: 'eq', text:
"@Resources.Resource.Equal" }, { oper: 'ne', text:
"@Resources.Resource.NotEqual" },
            { oper: 'lt', text:
"@Resources.Resource.Less" }, { oper: 'le', text:
"@Resources.Resource.LessOrEqual" },
            { oper: 'gt', text:
"@Resources.Resource.Greater" }, { oper: 'ge', text:
"@Resources.Resource.GreaterOrEqual" },
            { oper: 'nu', text:
"@Resources.Resource.IsNull" }, { oper: 'nn', text:
"@Resources.Resource.IsNotNull" }],
        groupOps: [{ op: "AND", text:
"@Resources.Resource.All" }, { op: "OR", text:
"@Resources.Resource.Any" }],

sopt: ['eq', 'ne', 'lt', 'le', 'gt', 'ge', 'nu', 'nn'],
        width: 700
    }, {
        caption:
"@Resources.Resource.ViewCaption",
        bClose: "@Resources.Resource.ViewClose",
        width: 1000
    });
</script>
}

```

```

<h2>@ViewBag.Title</h2>

<table id="jqg"></table>
<div id="jpager"></div>

```

@model Дипсертация.Models.SettingsViewModel

```

@{
    ViewBag.Title = Resources.Resource.Settings;
}

<script type="text/javascript">
function Validate()
{
    var container, inputs, counter;
    container =
document.getElementById('CheckBoxList1');
    inputs = container.getElementsByTagName('input');
    counter = 0;
    for (var i = 0; i < inputs.length; i++) {
        if (inputs[i].checked) {
            counter++;
        }
    }
    if (counter == 0) {
        alert("@Resources.Resource.AlertMetOpt");
        return false;
    }

    container =
document.getElementById('CheckBoxList2');
    inputs = container.getElementsByTagName('input');
    counter = 0;
    for (var i = 0; i < inputs.length; i++) {

```

```

        if (inputs[i].checked) {
            counter++;
        }
    }
    if (counter == 0) {
        alert("@Resources.Resource.AlertZeroLim");
        return false;
    }
    if (counter == inputs.length) {
        alert("@Resources.Resource.AlertManyLim" + "
" + (inputs.length-1) + ".");
        return false;
    }

    container =
document.getElementById('RadioButtonList');
    inputs = container.getElementsByTagName('input');
    counter = 0;
    for (var i = 0; i < inputs.length; i++) {
        if (inputs[i].checked) {
            counter++;
        }
    }
    if (counter == 0) {
        alert("@Resources.Resource.AlertCritOpt");
        return false;
    }

    return true;
}

```



```

}

function RadioClick(s)
{
    var container =
document.getElementById('CheckBoxList2');
    var inputs =
container.getElementsByTagName('input');
    for (var i = 0; i < inputs.length; i++) {
        if (s == inputs[i].id) {
            inputs[i].checked = false;
            inputs[i].disabled = true;
        }
        else {
            inputs[i].disabled = false;
        }
    }
}

function ready() {
    var container1, inputs1, container2, inputs2;
    container1 =
document.getElementById('CheckBoxList2');
    inputs1 =
container1.getElementsByTagName('input');
    container2 =
document.getElementById('RadioButtonList');
    inputs2 =
container2.getElementsByTagName('input');
    for (var i = 0; i < inputs1.length; i++) {
        for (var j = 0; j < inputs2.length; j++) {
            if ((inputs1[i].id == inputs2[j].id) &&
(inputs2[j].checked)) {
                inputs1[i].checked = false;
                inputs1[i].disabled = true;
            }
        }
    }
}

document.addEventListener("DOMContentLoaded", ready);
</script>

<h2>@ViewBag.Title</h2>

<p class="text-info">@ViewBag.StatusMessage</p>

<div class="container">
    @using (Html.BeginForm("Settings", "Home"))
    {
        <div class="row">
            <div class="col-md-4" id="CheckBoxList1">
                <h3>@Resources.Resource.MetOpt:</h3>
                @foreach (var met in
Model.MethodsList)
                {
                    <div class="checkbox">
                        <input type="checkbox"
                            name="SelectedMethods"

```

```

                            value="@met.Value"
                            id="@met.Value"
                        @if (met.Selected) {
                            <text> checked='checked' </text> } />
                        @met.Text
                    }
                }
            </div>
            <div class="col-md-4" id="CheckBoxList2">
                <h3>@Resources.Resource.Limitations:</h3>
                @foreach (var lim in
Model.LimitationsList)
                {
                    <div class="checkbox">
                        <input type="checkbox"

name="SelectedLimitations"
                            value="@lim.Value"
                            id="@lim.Value"
                        @if (lim.Selected) {
                            <text> checked='checked' </text> } />
                        @lim.Text
                    }
                }
            </div>
            <div class="col-md-4"
id="RadioButtonList">
                <h3>@Resources.Resource.CritOpt:</h3>
                @foreach (var cri in
Model.CriteriaList)
                {
                    <div class="radio">
                        <input type="radio"

name="SelectedCriteria"
                            value="@cri.Value"
                            id="@cri.Value"

onchange="RadioClick(this.id)"
                        @if (cri.Selected) {
                            <text> checked='checked' </text> } />
                        @cri.Text
                    }
                }
            </div>
            <br/>
            <div class="row">
                <div class="form-group text-center">
                    <input type="submit" class="btn btn-
primary" value="@Resources.Resource.Save"
onclick="return Validate()"/>
                </div>
            </div>
        }
    }
</div>

```

```

@model Дипсертация.Models.ChangePasswordViewModel
@{
    ViewBag.Title =
Resources.Resource.ChangePasswordHeader;
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("ChangePassword", "Manage",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))
{
    @Html.AntiForgeryToken()
    <hr />

```

```

    @Html.ValidationSummary("", new { @class = "text-
danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.OldPassword, new {
@class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.OldPassword, new
{ @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(m => m.NewPassword, new {
@class = "col-md-2 control-label" })
        <div class="col-md-10">

```

```

        @Html.PasswordFor(m => m.NewPassword, new
    { @class = "form-control" })
    </div>
</div>
<div class="form-group">
    @Html.LabelFor(m => m.ConfirmPassword, new {
@class = "col-md-2 control-label" })
    <div class="col-md-10">
        @Html.PasswordFor(m => m.ConfirmPassword,
new { @class = "form-control" })
    </div>
</div>
<div class="form-group">

```

```

-----
@model Диссертация.Models.EditAccountModel
@{
    ViewBag.Title =
Resources.Resource.EditAccountHeader;
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("EditAccount", "Manage",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))
{
    @Html.AntiForgeryToken()
    <hr />
    @Html.ValidationSummary("", new { @class = "text-
danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.DateBirth, new { @class
= "col-md-2 control-label" })
    </div>

```

```

-----
@model Диссертация.Models.IndexViewModel
@{
    ViewBag.Title = Resources.Resource.YourAccount;
}

<h2>@ViewBag.Title</h2>

<p class="text-success">@ViewBag.StatusMessage</p>
<div>
    <hr />
    <dl class="dl-horizontal">
        <dt>@Resources.Resource.Password:</dt>
        <dd>
            [
                @if (Model.HasPassword)
                {
                    <a
href="/EngineOpt/Manage/ChangePassword" class="btn-
link">@Resources.Resource.ChangePassword</a>
                }
                else
                {

```

```

-----
@model Диссертация.Models.SetPasswordViewModel
@{
    ViewBag.Title =
Resources.Resource.SetPasswordHeader;
}

<h2>@ViewBag.Title</h2>

@using (Html.BeginForm("SetPassword", "Manage",
FormMethod.Post, new { @class = "form-horizontal",
role = "form" }))
{
    @Html.AntiForgeryToken()

```

```

        <div class="col-md-offset-2 col-md-10">
            <input type="submit"
value="@Resources.Resource.Change" class="btn btn-
primary" />
        </div>
    </div>
}
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

        <div class="col-md-10">
            @Html.TextBoxFor(m => m.DateBirth, new {
@class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <input type="submit"
value="@Resources.Resource.Change" class="btn btn-
primary" />
        </div>
    </div>
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

        <a
href="/EngineOpt/Manage/SetPassword" class="btn-
link">@Resources.Resource.CreatePassword</a>
    }
}
</dd>
<dt>@Resources.Resource.PersonalData:</dt>
<dd>
    [ <a href="/EngineOpt/Manage/EditAccount"
class="btn-link">@Resources.Resource.EditAccount</a>
]
</dd>
    @if (Model.DateBirth != null)
    {
        <dt>@Resources.Resource.DateBirth:</dt>
        <dd>
            @Model.DateBirth.Value.ToShortDateString()
        </dd>
    }
</dl>
</div>

```

```

        <hr />
        @Html.ValidationSummary("", new { @class = "text-
danger" })
    <div class="form-group">
        @Html.LabelFor(m => m.NewPassword, new {
@class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.NewPassword, new
{ @class = "form-control" })
        </div>
    </div>
    <div class="form-group">

```

```

        @Html.LabelFor(m => m.ConfirmPassword, new {
@class = "col-md-2 control-label" })
        <div class="col-md-10">
            @Html.PasswordFor(m => m.ConfirmPassword,
new { @class = "form-control" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">

```

```

        <input type="submit"
value="@Resources.Resource.Create" class="btn btn-
primary" />
    </div>
}
@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}

```

```

<ul class="nav navbar-nav">
    <li class="dropdown">
        <a href="#" class="dropdown-toggle" data-
toggle="dropdown" role="button" aria-
expanded="false">
            @Resources.Resource.ChooseLang <span
class="caret"></span>
        </a>
        <ul class="dropdown-menu" role="menu"
name="lang">

```

```

        <li><a
href="~/Home/ChangeCulture?lang=ru">Русский</a></li>
        <li><a
href="~/Home/ChangeCulture?lang=uk">Українська</a></li>
        <li><a
href="~/Home/ChangeCulture?lang=en">English</a></li>
    </ul>
</li>
</ul>

```

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>EngineOpt</title>
    @Styles.Render("~/Content/css")
    @Styles.Render("~/Content/MyStyles.css")
    @Scripts.Render("~/bundles/modernizr")
    @RenderSection("styles", required: false)
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-
top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-
toggle" data-toggle="collapse" data-target=".navbar-
collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("EngineOpt",
"MainMenu", "Home", new { area = "" }, new { @class =
"navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav">

```

```

<li>@Html.ActionLink(Resources.Resource.MainMenu,
"MainMenu", "Home")</li>
<li>@Html.ActionLink(Resources.Resource.Settings,
"Settings", "Home")</li>
<li>@Html.ActionLink(Resources.Resource.Results,
"Results", "Home")</li>
<li>@Html.ActionLink(Resources.Resource.About,
"About", "Home")</li>
    </ul>
    @Html.Partial("_LangPartial")
    @Html.Partial("_LoginPartial")
</div>
</div>
<div class="container body-content">
    @RenderBody()
    <hr />
    <footer>
        <p>@DateTime.Today.ToShortDateString() -
EngineOpt</p>
    </footer>
</div>
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>

```

```

@using Microsoft.AspNet.Identity
@if (Request.IsAuthenticated)
{
    using (Html.BeginForm("LogOff", "Account",
FormMethod.Post, new { id = "logoutForm", @class =
"navbar-right" }))
    {
        @Html.AntiForgeryToken()

        <ul class="nav navbar-nav navbar-right">
            <li>
                @Html.ActionLink(Resources.Resource.Hello
+ ", " + User.Identity.GetUserName(), "Index",
"Manage", routeValues: null, htmlAttributes: null)

```

```

        </li>
        <li><a
href="javascript:document.getElementById('logoutForm')
.submit()">@Resources.Resource.Exit</a></li>
    </ul>
}
else
{
    <ul class="nav navbar-nav navbar-right">
        <li>@Html.ActionLink(Resources.Resource.Register,
"Register", "Account", routeValues: null,
htmlAttributes: null)</li>

```

```
<li>@Html.ActionLink(Resources.Resource.LoginHeader,    }    </ul>
"Login", "Account", routeValues: null,
htmlAttributes: null)</li>
```